Cleveland State University
Department of Electrical and Computer Engineering

FINAL REPORT

For

Robust Timing Synchronization in Aeronautical Mobile Communication Systems

Total Period of Research: May 21, 2001 to August 20, 2004

Written by

Fuqin Xiong, Ph. D. Principal Investigator, Professor
Stanley Pinchak, Graduate Assistant

Department of Electrical and Computer Engineering, Cleveland State University
Cleveland, Ohio, 44115

Submitted to
NASA Glenn Research Center

Grant NASA/NAG 3-2619 Final Report, September 16, 2004

Table of Contents

ACKNOWLEDGMENT

This research was in close collaboration with Mr. Kue Chun of NASA Glenn Research Center, Digital Communications Technology Branch. The grant was also managed Mr. Kue Chun. Part of analysis and all simulations were carried out by graduate assistant, Stanley Pinchak, of the Department of Electrical and Computer Engineering, Cleveland State University. The PI would like to thank all above individuals for their contributions to the completion of this project.

Summary

This work details a study of robust synchronization schemes suitable for satellite to mobile aeronautical applications. A new scheme, the Modified Sliding Window Synchronizer (MSWS), is devised and compared with existing schemes, including the traditional Early-Late Gate Synchronizer (ELGS), the Gardner Zero-Crossing Detector (GZCD), and the Sliding Window Synchronizer (SWS). Performance of the synchronization schemes is evaluated by a set of metrics that indicate performance in digital communications systems. The metrics are convergence time, mean square phase error (or root mean-square phase error), lowest SNR for locking, initial frequency offset performance, midstream frequency offset performance, and system complexity.

The performance of the synchronizers is evaluated by means of Matlab simulation models. A simulation platform is devised to model the satellite to mobile aeronautical channel, consisting of a Quadrature Phase Shift Keying modulator, an additive white Gaussian noise channel, and a demodulator front end. Simulation results show that the MSWS provides the most robust performance at the cost of system complexity. The GZCD provides a good tradeoff between robustness and system complexity for communication systems that require high symbol rates or low overall system costs. The ELGS has a high system complexity despite its average performance. Overall, the SWS, originally designed for multi-carrier systems, performs very poorly in single-carrier communications systems. The following table, which is later Table 5.1 in Section 5, provides a ranking of each of the synchronization schemes in terms of the metrics set forth in Section 4.1. Details of comparison are given in Section 5.

Robust Synchronization Performance Table

Synchronizer	Convergence (Number of Symbols)	Root Mean-Square Phase Error[1]	Lowest SNR for Locking[2]	Initial Frequency Offset	Midstream Frequency Offset	Frequency Offset @ Lowest SNR	Complexity
SWS 8Tap	350	0.0261T	3dB	±50 ppm	±50 ppm	N/A	7
SWS 4Tap	350	0.0114T	3dB	N/A	N/A	N/A	5
MSWS 8Tap	10	0.0033T	-3dB	±100 ppm	±100 ppm	±100 ppm	6
MSWS 4Tap	10	0.0122T	0dB	±100 ppm	±50 ppm	±100 ppm	4
ELGS 8Tap	50	0.0122T	3dB	±100 ppm	±100 ppm	±100 ppm	3
ELGS 4Tap	50	0.0412T	4dB	±100 ppm	±100 ppm	±100 ppm	2
GZCD	50	0.0144T	3dB	±100 ppm	±100 ppm	±100 ppm	1

[1] T=symbol period, [2] E_b/N_0

Based on the results presented in the table, it is safe to say that the most robust synchronization scheme examined in this work is the high-sample-rate Modified Sliding Window Synchronizer. A close second is its low-sample-rate cousin. The tradeoff between complexity and lowest mean-square phase error determines the rankings of the Gardner Zero-Crossing Detector and both versions of the Early-Late Gate Synchronizer. The least robust models are the high and low-sample-rate Sliding Window Synchronizers. Consequently, the recommended replacement synchronizer for NASA's Advanced Air Transportation Technologies mobile aeronautical communications system is the high-sample-rate Modified Sliding Window Synchronizer. By incorporating this synchronizer into their system, NASA can be assured that their system will be operational in extremely adverse conditions. The quick convergence time of the MSWS should allow the use of high-level protocols. However, if NASA feels that reduced system complexity is the most important aspect of their replacement synchronizer, the Gardner Zero-Crossing Detector would be the best choice.

1. Introduction

Robust synchronization is a crucial technique for a modem to function properly. The modem of the AATT mobile Ku-Band earth station developed by NASA Glenn Research Center is an EB200 spread spectrum modem manufactured by L-3 Corporation [1]. Field test results show that the AATT mobile Ku-Band earth station has experienced communication outages during tests, when the mobile unit passes under highway overpasses or shadowed by other obstacles [2]. One of the possible causes of the outages is the loss of symbol timing synchronization. This project is to investigate robust symbol timing synchronization techniques and to recommend the suitable one for use in the mobile satellite communications.

The Sliding Window Synchronizer (SWS) as proposed by Arafat Al-Dweik [3] shows resilience to low signal to noise ratio (SNR). We first focused our effort on the analysis and simulation of this scheme. Further research into this synchronizer showed that this scheme is best suitable for a multi-carrier system, like OFDM system. For single-carrier communication systems, some modifications are needed. As a result, a Modified Sliding Window Synchronizer (MSWS) was proposed. To judge the merit of the proposed MSWS, besides the original SWS, two other popular symbol synchronizers were selected for comparison. First, the popular Early-Late Gate Synchronizer (ELGS) [7] was chosen to serve as a baseline against which to compare the others over the course of our evaluation. Second, the Gardner's Zero-Crossing Detector (GZCD) as proposed by F. Gardner was also chosen for comparison due to its high efficiency [4]. Each of these synchronization techniques falls into the non-data-aided, ad hoc category. While this was not intentional, the selection is appropriate given that low SNR capabilities are common to many non-data-aided synchronizers and that most synchronizers in use are of the ad hoc variety. In-depth description and analysis of each of these techniques is contained in Section 2

Performance of the synchronization schemes is evaluated by a set of metrics that indicate performance in digital communications systems. The metrics are convergence time, mean-square phase error, lowest SNR for locking, initial frequency offset performance, midstream frequency offset performance, and system complexity.

The performance of the synchronizers is evaluated by means of Matlab simulation models. A simulation platform is devised to model the satellite to mobile aeronautical channel, consisting of a QPSK modulator, an AWGN channel, and a demodulator front end. The Matlab models of the robust synchronization candidates are derived from their published descriptions in the case of the SWS [3] and GZCD [4]. The simulation model of the ELGS is derived from the description in reference [7]. The simulation model of the MSWS is newly proposed.

Simulation results show that the MSWS provides the most robust performance at the cost of system complexity. The GZCD provides a good tradeoff between robustness and system complexity for communication systems that require high symbol rates or low overall system costs. The ELGS has a high system complexity despite its average performance. Overall, the SWS, originally designed for multi-carrier systems, performs very poorly in single-carrier communications systems.

This report is based on the material from a published conference paper [6] and the master's thesis of graduate assistant, Stanley Pinchak, under the supervision of this principal investigator [8]. This report consists of five sections: Introduction, SWS and Modified SWS, ELGS and GZCD, Performance Evaluation, Conclusions and Recommendations.

2. Sliding Window Synchronizer and Modified Sliding Window Synchronizer

The Sliding Window Synchronizer is an ad hoc technique first proposed by Arafat Al-Dweik in 2000 for use in multi-carrier Orthogonal Frequency Division Multiplexing (OFDM) communications systems [3]. Published results showed very good performance even in environments with a SNR as low as 0 dB. However, adapting the SWS to the single carrier communication system proved to be difficult. Initial results showed rather poor performance, and various combinations of changes to the loop filter and gains did not significantly improve this. Despite these poor results, in the interest of furthering the understanding of the SWS, the original scheme is described and analyzed. The high-sample-rate required by the original scheme encouraged the creation of a low-sample-rate version. The original scheme proposed the use of eight samples per symbol, whereas the low-sample-rate technique operates on four samples per symbol. The performance of the low-sample-rate SWS in simulations could not be improved despite repeated tuning.

2.1 Sliding Window Synchronizer (SWS)

The original Sliding Window Synchronizer is shown in Figure 2.1. The first step of the SWS synchronization process begins with a baseband symbol stream. The incoming baseband signal is sampled in both the In-Phase (I) and Quadrature-Phase (Q) channels. The samplers are controlled by the voltage-controlled oscillator (VCO) or numeric controlled oscillator (NCO). These samplers operate at eight times the frequency of the incoming symbol rate.

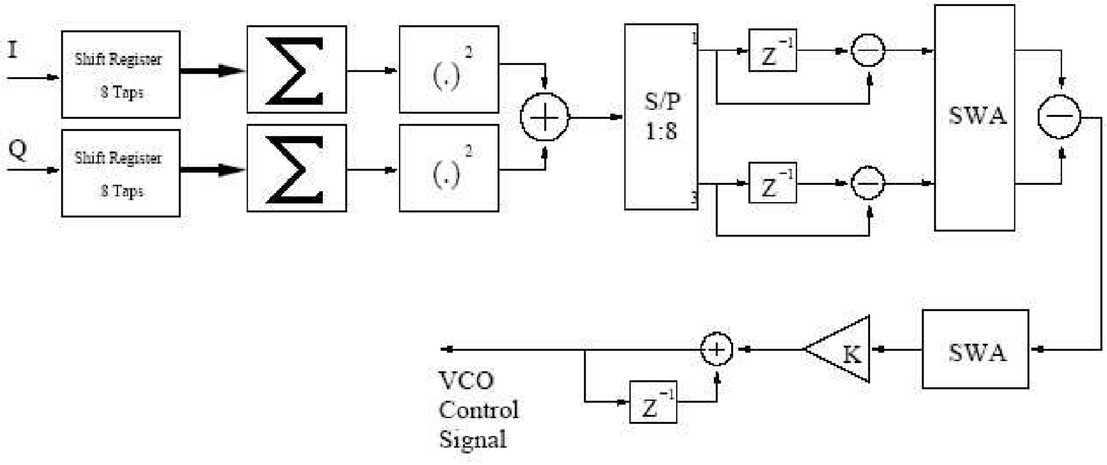


Figure 2.1: Sliding Window Synchronizer Block Diagram

Every sample time, the contents of each shift register are summed and the resulting summations are squared. Immediately, the two sum-squared magnitude values from the I and Q channels are added together and fed into a serial-to-parallel converter (S/P) of width eight taps. The output of the S/P is sampled once per symbol period by a clock-divided version of the VCO or NCO. The outputs of the first and third taps of the S/P are then subtracted from the first and third taps of S/P from the previous symbol. The two difference signals are each fed into a sliding window averager (SWA) that is 128 taps long. The sum of the tap 3 SWA is subtracted from the sum of the tap 1 SWA and the result is sent to another SWA, which is also 128 taps long.

The output of the difference SWA is then sent to the loop filter and integrator. The output of the integrator is used to control a VCO or NCO with a free-running frequency equal to eight times the incoming symbol rate. The output of the VCO or NCO is sent to a clock divider to create the symbol strobe signal. This symbol strobe is used to control the S/P and can be compared to the incoming symbol edge for the purposes of performance evaluation.

2.2 Modified Sliding Window Synchronizer (SWS)

We have modified the structure and the algorithm proposed in [3], the block diagram is given in Figure 2.2. The operation of the MSWS is similar to that of SWS up to the output of the S/P block. The control algorithm of the synchronizer is modified as shown in the figure. Instead of comparing tap 1 and 3, a maximum tap selector is used. Details of this algorithm will be described later.

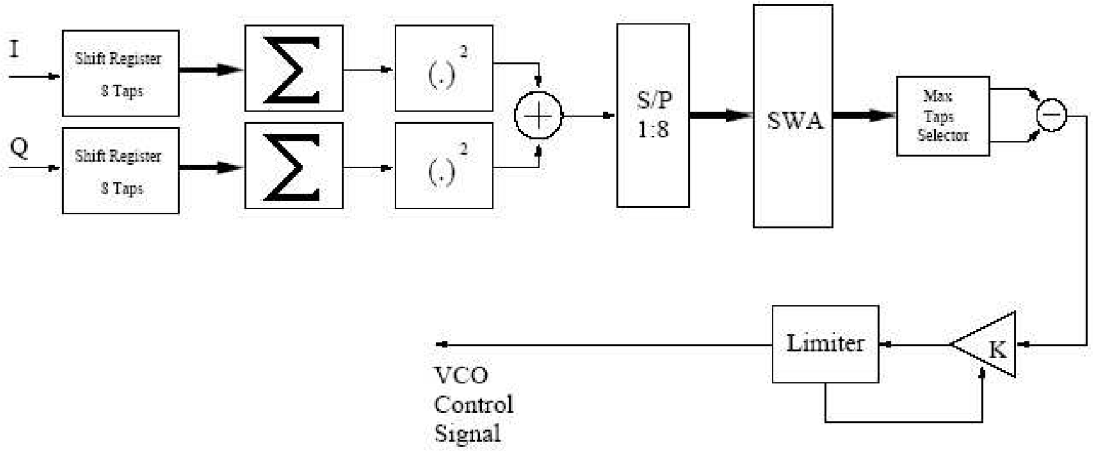


Figure 2.2: Modified sliding window symbol synchronizer block diagram

If we denote the output of the S/P in Figure 2.1 and 2.2 as W(n), where n is time index, W(n) is in effect a digital phase detector. The SWA computes the average of the W(n) over a sufficient time length to produce the phase detector characteristic W(L,K,a), where K is number of samples in the sliding window, L is the symbol timing offset (delay or advance) in number of samples between the sliding window and the incoming data symbol timing, a is the ratio of the correct sampling frequency over the local sampling frequency. The control algorithm computes three points on the phase detector characteristic and compares them to decide what control signal (+ or -) is to send to the VCO.

Since control algorithm is based on it, in the following we will derive the phase detector characteristic W(L,K,a).

2.3. Sampled MPSK signals

On the entire time axis, the I-channel and Q-channel components of an MPSK signal in baseband can be written as

$$I(t) = \sum_{k=-\infty}^{\infty} I_k p(t-kT) + n(t), \quad -\infty < t < \infty$$

$$Q(t) = \sum_{k=-\infty}^{\infty} Q_k p(t-kT) + n(t), \quad -\infty < t < \infty$$

where I_k and Q_k are the amplitude of the I-channel and Q-channel signal, respectively. p(t) is the pulse shape. Here we assume p(t) is a rectangular pulse in [0,T] , where T is the symbol period. n(t) is the white Gaussian noise with two-sided power spectral density (PSD) of No/2. I_k and Q_k are determined by the phase θ_k of the signal in the kth period:

$$I_k = A\cos\theta_k, \qquad Q_k = A\sin\theta_k \tag{1}$$

where A is the amplitude of the MPSK signal. In the following, we set A=1 without loss of generality. The phase θ_k is determined by $\log_2 M$ data bits represented by the modulated signal: the symbol; and $\theta_k \in \{\theta_i\}_1^M$, where θ_i are evenly spaced angles on the signal constellation. The expression for θ_i is

$$\theta_i = \frac{(M-4i+2)\pi}{2M}, \quad i = 1,2,\ldots,M \tag{2}$$

When sampled, I(t) and Q(t) become $I(nT_s)=I(n)$ and $Q(nT_s)=Q(n)$, where T_s is the sampling period. Now let us observe K samples in a sliding window of the length of the symbol period T. The outputs of the summers in Figure 2.1 are the sums of these K samples

$$S_I(n) = \sum_{k=0}^{K-1} I(n-k), \qquad S_Q(n) = \sum_{k=0}^{K-1} Q(n-k)$$

and the output of the adder is

$$W(n) = S_I^2(n) + S_Q^2(n)$$

2.4. Phase Detector Characteristic (the W(L,K,a) curve)

To derive the phase detector characteristic, we need to examine all possible cases in terms of the relationship between incoming data symbol clock, the local sampling clock and the symbol clock, ignoring the noise. To illustrate these different cases, we use 8PSK as an example (Figure 2.3), where samples of all 8 possible 8PSK symbols are shown, with 8 samples per symbol. The local symbol clock is derived from the sampling clock by frequency divider. The sliding window length is defined as the number of samples in the window. The sliding window duration thus varies with the sampling frequency or indirectly with the clock frequency.

Case 1: the sampling frequency is correct and symbol timing is perfect (Figure 2.3, case 1). In this case, the K samples have exactly the same amplitude, $A\cos\theta_k$ and $A\sin\theta_k$, in I- and Q-channel, respectively. Then

$$S_I(n) = KA\cos\theta_k, \qquad S_Q(n) = KA\sin\theta_k$$

$$W(n) = K^2A^2$$

This is the maximum value of W(n), and it is proportional to the baseband symbol energy (the baseband symbol energy is $TA^2/2$).

Case 2: the sampling frequency is correct (thus the symbol clock frequency is the same as the data symbol rate), but the symbol timing is delayed by L samples. In this case, the sliding the window covers K-L samples of the current symbol and L samples of the next symbol (Figure 2.3, case 2).

Case 3: the sampling frequency is correct, but the symbol timing is advanced by L samples. In this case, sliding the window covers K-L samples of the current symbol and L samples of the previous symbol (Figure 2.3, case 3).

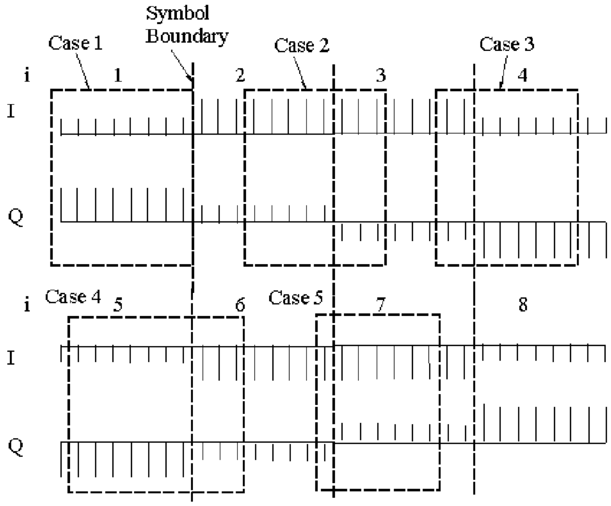
Case 4: the sampling frequency is lower than the correct frequency (thus local symbol clock frequency is lower than the data symbol rate). In this case, the number of samples in the sliding window is still K since the clock frequency is derived from the sampling frequency by dividing it by K. However, these K samples are not from only one symbol. They may be from two or more symbols depending on how low the clock frequency is in comparison with the data symbol rate. Usually the clock frequency is not too far away from the data symbol rate. Thus a reasonable assumption is that the clock frequency is at least 1/2 of the symbol rate. Thus, these K symbols are at most from 3 symbols.

Case 5: the sampling frequency is higher than the correct frequency (thus symbol clock frequency is higher than the data symbol rate). In this case, the number of samples in the sliding window is still K. The samples could be from one symbol or at most two symbols.

In both case 4 and case 5, the clock phase is in general different from the data symbol-timing phase. Their starting edges may coincide occasionally, however the probability for them to be exactly the same is zero. Even if their starting edges indeed coincide, their ending edges will be at different times since their frequencies are different. If the local clock is delayed by L samples, the case is called Case 4d or 5d. If the local clock is advanced by L samples, the case is called Case 4a or 5a. Now we try to investigate the behavior of phase detector.

2.4.1. Same Frequency, Different Phase

In cases 2 and 3, the local clock has the same frequency but different phase with respect to the data symbol timing. It will be seen shortly that W(L, K) has the same behavior in these two cases. If the clock frequency is not exactly the same as that of the data symbol timing but the difference in clock period is within 1/2 sample period, it is easy to see that cases 2 and 3 still apply to the situation.



8PSK signal samples:
|high|=cos(22.5 degree)=sin(67.5 degree)=0.9238
|low|=sin(22.5 degree)= cos(67.5 degree)=0.3826

Figure 2.3: Illustration of 5 cases using 8PSK signals as an example.

We start the discussion with case 2. Assume there is a delay of L samples in symbol, the sum is

$$W(n,L,K,i,j)=\left[\sum_{k=0}^{L-1} I_j(n-k)+\sum_{k=L}^{K-1} I_i(n-k)\right]^2 + \left[\sum_{k=0}^{L-1} Q_j(n-k)+\sum_{k=L}^{K-1} Q_i(n-k)\right]^2$$

where i and j indicate current data symbol i and the next data symbol j. It is easy to see that the value of W(n, L, K, I, j) does not really depend on time index n, provided that data are a stationary, ergodic random process, which is usually the case. Rather it is a function of the symbol pattern, consisting of the current data symbol i and the next data symbol j, in addition to L and K. There are L samples of symbol j and K-L samples of symbol i. Since for MPSK, samples within a symbol have the same amplitude, we can write

$$W(L,K,i,j)=[LI_j+(K-L)I_i]^2+[LQ_j+(K-L)Q_i]^2 \tag{3}$$

where time index n in I- and Q-channel signals are suppressed. In fact this formula also can be used for QAM since its samples within a symbol also have the same amplitude.

For MPSK, there are M^2 two-symbol patterns. If we index the symbols as 1, 2, ... M, the two-symbol patterns' indexes form an M×M matrix as

$$\begin{bmatrix} 11 & 12 & \dots & 1M \\ 21 & 22 & \dots & 2M \\ \vdots & \vdots & \vdots & \vdots \\ M1 & M2 & \dots & MM \end{bmatrix}$$

Inspecting Figure 2.3, it can be easily verified that the value of W(L, K, i, j) for pattern ij in case 2 for a delay of L is equal to that of pattern ji in case 3 for an advance of L. It can be seen from the matrix that for a pattern ij, there is always a symmetrical pattern ji. Thus the statistical average of W(L, K, i, j) over all possible patterns is the same for case 2 and case 3. It is therefore suffice to study the behavior of case 2. Thus the average of W(L, K, i, j) is only a function of L and K denoted as W(L, K).

$$W(L,K)=\frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} W(L,K,i,j) \tag{4}$$

With the aid of a computer program (MathCAD), the values of W(L, K, i, j) and W(L,K) can be calculated. It turns out that the values of W(L, K, i, j) is the same as that of W(L, K, j, i), that is, the matrix of W(L, K, i, j) is symmetrical about the diagonal. This can be proved as follows.

Expanding (3) and using the fact that

$$I_i^2+Q_i^2=1 \tag{5}$$

we have

$$W(L,K,i,j)=L^2+(K-L)^2+2L(K-L)(I_jI_i+Q_jQ_i) \tag{6}$$

The above expression is invariant when indexes i and j are exchanged, thus the proof. Further, from above or (3), we can see that the values on the diagonal (i = j) are K^2, which is the case that two adjacent symbols are the same so that W(L, K, i, j) is maximum regardless the value of delay L.

Computer calculation also revealed that the values of W(L,K) for MPSK schemes are the same for any modulation order M, provided L and K are fixed. Analytically, this can be proved as follows. Using (5) in (4),

$$W(L,K)=\frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}W(L,K,i,j)=L^2+(K-L)^2+\frac{2L(K-L)}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}(I_iI_j+Q_iQ_j)$$

The sum in the last term can be evaluated using (1) and (2) as follows

$$\sum_{i=1}^{M}\sum_{j=1}^{M}(I_iI_j+Q_iQ_j)=\sum_{i=1}^{M}\sum_{j=1}^{M}(\cos\theta_i\cos\theta_j+\sin\theta_i\sin\theta_j)$$

$$=\sum_{i=1}^{M}\sum_{j=1}^{M}\cos(\theta_i-\theta_j)=\sum_{i=1}^{M}\sum_{j=1}^{M}\cos\left(\frac{2\pi(j-i)}{M}\right)$$

$$=\sum_{i=1}^{M}\cos\left(\frac{2\pi(1-i)}{M}\right)+\sum_{i=1}^{M}\cos\left(\frac{2\pi(2-i)}{M}\right)+\ldots+\sum_{i=1}^{M}\cos\left(\frac{2\pi(M-i)}{M}\right)$$

In the above each sub-sum is equal to zero since the phases of the cosine terms in the sub-sum just completely match to all phases of the constellation points. (The above term is also first revealed by MathCAD as zero). Thus

$$\sum_{i=1}^{M}\sum_{j=1}^{M}(I_iI_j+Q_iQ_j)=0 \tag{7}$$

and

$$W(L,K)=L^2+(K-L)^2 \tag{8}$$

which is only a function of L and K, regardless of modulation order M. It reaches maximum at L=0 or L=K:

$$\max W(L,K)=K^2$$

That is, it is the maximum when synchronism is reached. Figure 2.4 is the plots of (8).

Note that we previously proved that W(L,K) of case 3 is the same as that of case 2. That is, the expression of W(L,K) in (7) can be used for case 2, where L is the delay of the sliding window, and for case 3, where L is the advance of the sliding window.

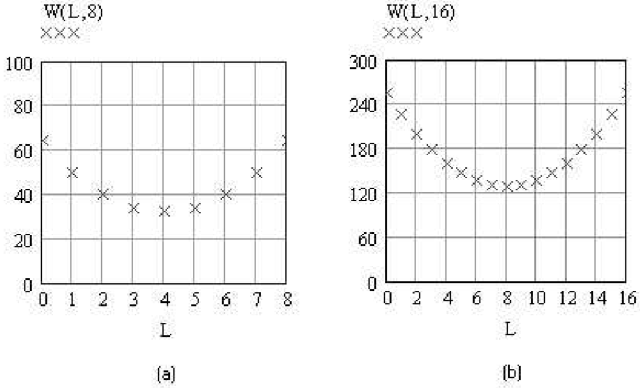


Figure 2.4: Average squared sum W(L,K) for cases 2 and 3. (a) K=8, (b) K=16. Note that only points at integer L exist.

2.4.2. Different Frequency and Phase

In cases 4 and 5, the local clock has different frequency from the data symbol timing, consequently the phases are different in general.

In case 4, the local sampling frequency f_l is lower than the correct frequency f_s , or equivalently, the sampling period T_l is longer than the correct sampling period T_s. That is

$$f_l=f_s/a \quad or \quad T_l=aT_s, \quad a\geq 1$$

where a is a positive number slightly greater than 1. Then, the window length, which is a period of the clock, is

$$KT_l = aKT_s$$

In case 4d, the window is delayed by L samples with a new sampling period T_l (Figure 2.5). We assume that $LT_l + (a-1)KT_s \le KT_s$. It means the sum of the extra window length and the delay is no larger than a correct symbol period; this is a reasonable assumption since we do not expect that the clock frequency is off too much. This leads to the relation between a and L as

$$a \le \frac{2K}{L+K} \text{ or } L \le \left(\frac{2}{a}-1\right)K \tag{9}$$

Part of the window length that covers the current data symbol (symbol i) is $(KT_s - LT_l) = (K - La)T_s$ and that covers the next data symbol (symbol j) is $KT_l - (KT_s - LT_l) = (Ka - K + La)T_s$. Thus the number of samples with T_l of the current symbol is

$$M_I = \left\lfloor \frac{(K-La)T_s}{T_l} + 0.5 \right\rfloor = \left\lfloor \frac{K}{a} - L + 0.5 \right\rfloor$$

where $\lfloor x \rfloor$ indicates the integer floor of x. The above expression is true when sampling instant is in the first half of the sampling period, which is usually the case. Similarly, in the next symbol (symbol j), the number of samples is

$$M_J = \left\lfloor \frac{(Ka-K+La)T_s}{T_l} + 0.5 \right\rfloor = \left\lfloor K - \frac{K}{a} + L + 0.5 \right\rfloor$$

The total number of samples in the window is K, which should be and is indeed equal to M_I+M_J.

$$W(L,K,a,i,j) = [M_I I_i + M_J I_j]^2 + [M_I Q_i + M_J Q_j]^2 \tag{10}$$

Using (5) and (7), the statistical average is found to be

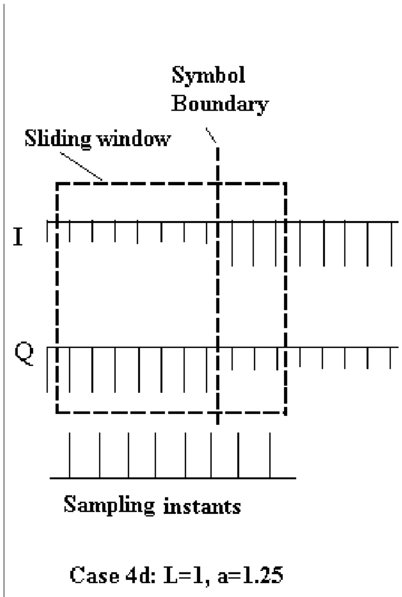


Figure 2.5: Case 4d: the sampling frequency is lower than correct and the window is delayed by LT$_l$.

$$W(L,K,a)=\frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}W(L,K,a,i,j)=M_I^2+M_J^2=\left\lfloor \frac{K}{a}-L+0.5\right\rfloor^2+\left\lfloor K-\frac{K}{a}+L+0.5\right\rfloor^2 \qquad (11)$$

The plots for K=8 and various a are shown in Figure 2.6. Some curves are completely or partially overlapped. For a fixed a the curves behave similar to case 2 and 3. Note that the condition given by (10) must be satisfied; so different curves for different a values terminated at different L values. There is one common feature; that is, the maximum is always at L=0. However, for certain a values (1.0,1.1,1.25,1.3), the maximum also appears at end of the curve; for others (1.05,1.15,1.2), the maximum appears only at the starting point (L=0). The ending point is not a maximum.

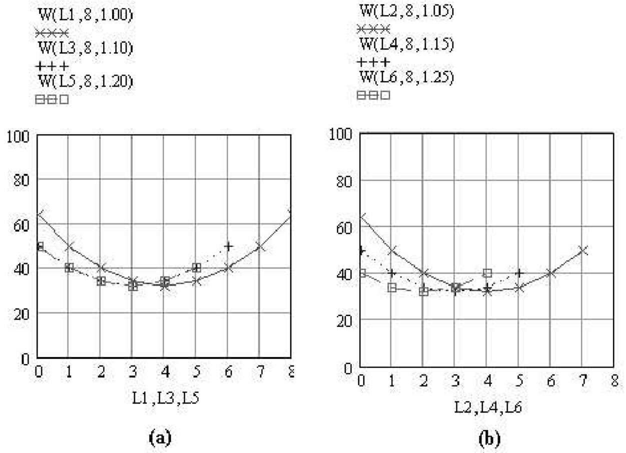


Figure 2.6: Average W(L,K,a) for Case 4d: (a) a=1,1.1,1.2, (b) a=1.05,1.15,1.25. Note that only points at L equal to integers exist. The lines show the trend of change.

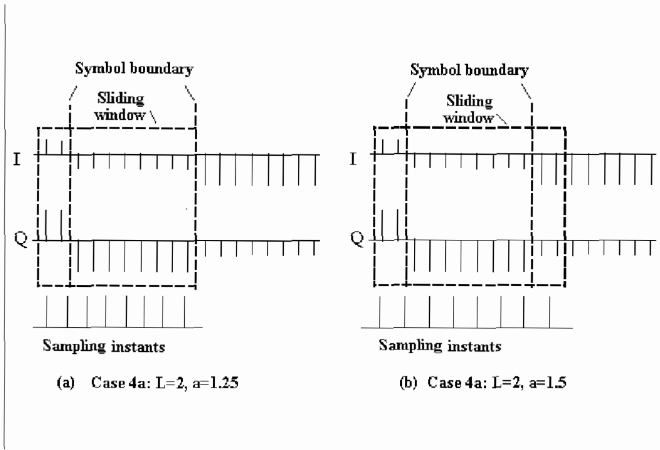
In case 4a, the window is advanced by LT_l. In the window, the number of the samples from the previous symbol is $M_K = L$. To calculate the number of the samples from the current symbol and next symbol, we must consider two situations. The first situation is that the remaining length of the window is entirely in the current symbol (Figure 2.7(a)). In this case, $(K-L)a \le K$, and the number of samples from the current symbol is $M_I = K - L$, and there are no samples from the next symbol. The second case is that the remaining length of the window is still longer than a symbol length (Figure 2.7(b)). In this case, $(K-L)a > K$, the number of samples from the current symbol is

$$M_I=\left\lfloor \frac{KT_s}{T_l}+0.5\right\rfloor=\left\lfloor \frac{K}{a}+0.5\right\rfloor$$

and the number of samples from the next symbol is $M_J=\left\lfloor \frac{KT_l-LT_l-KT_s}{T_l}+0.5\right\rfloor=\left\lfloor K-L-\frac{K}{a}+0.5\right\rfloor$

$$W(L,K,a,i,j,k)=\begin{cases}[M_K I_k+M_I I_i]^2+ \\ [M_K Q_k+M_I Q_i]^2, & if\,(K-L)a\le K \\ [M_K I_k+M_I I_i+M_J I_j]^2+ \\ [M_K Q_k+M_I Q_i+M_J Q_j]^2, & if\,(K-L)a> K\end{cases}$$

Using (5) and (7) the statistical average is found to be



(a) Case 4a: L=2, a=1.25

(b) Case 4a: L=2, a=1.5

Figure 2.7: Case4a: the sampling frequency is lower than correct, and the window is advanced by LT'–1". (a) the window covers the current symbol and part of the previous symbol. (b) the window covers the current symbol and parts of the previous and next symbol

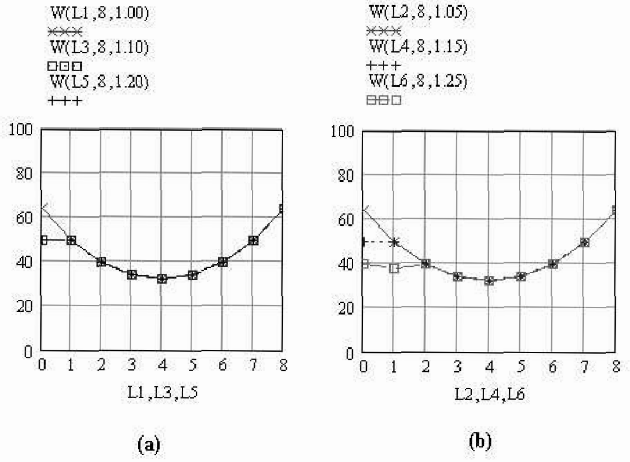


Figure 2.8: W(L,K,a) for Case 4a: (a) a=1,1.1,1.2, (b) a=1.05,1.15,1.25. Note that only points at L equal to integers exist. The lines show the trend of change.

$$W(L,K,a)=\frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M}W(L,K,a,i,j,k)=\begin{cases} M_K^2+M_l^2, & if\,(K-L)a\le K \\ M_K^2+M_l^2+M_j^2, & if\,(K-L)a>K \end{cases} \qquad (12)$$

$$=\begin{cases} L^2+(K-L)^2, & if\,(K-L)a\le K \\ L^2+\left\lfloor \frac{K}{a}+0.5\right\rfloor^2+\left\lfloor K-L-\frac{K}{a}+0.5\right\rfloor^2, & if\,(K-L)a>K \end{cases}$$

The plots are in Figure 2.8.

In case 5, the sampling frequency is higher than the correct frequency. The window is shorter than a symbol period. We assume that the sampling period is $T_l = aT_s, \quad a<1$

In case 5d, the clock is delayed by LT$_l$ and the window could or could not cover the next symbol, depending on L (Figure 2.9). If $LT_l \le KT_s - KT_l$ or $L \le K/a - K$, the window will not cover the next symbol, otherwise it will. When the window does not cover the next symbol (Figure 2.9(a)), the number of samples of the current symbol is K. When the window covers the current symbol and next symbol (Figure 2.9(b)), the number of samples of the current symbol is

$$M_l=\left\lfloor \frac{KT_s}{T_l}-L+0.5\right\rfloor=\left\lfloor \frac{K}{a}-L+0.5\right\rfloor$$

and the number of samples of the next symbol is

$$M_J = \left\lfloor \frac{KT_l + KT_l - KT_s}{T_l} + 0.5 \right\rfloor = \left\lfloor L + K - \frac{K}{a} + 0.5 \right\rfloor$$

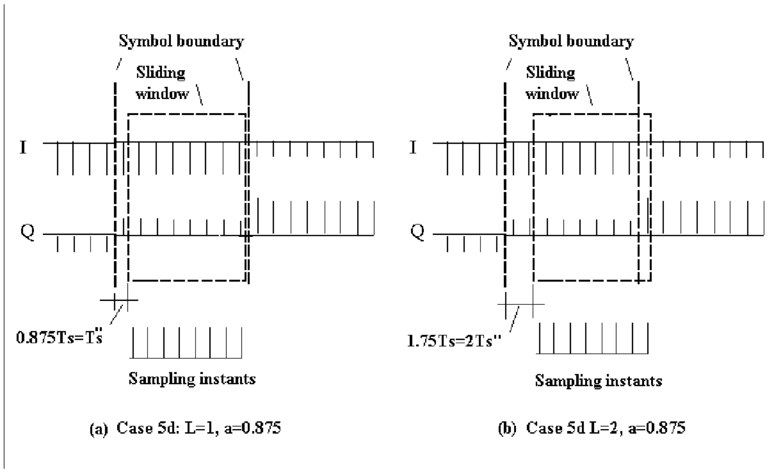
Thus

$$W(L,K,a,i,j) = \begin{cases} [KI_i]^2 + [KQ_i]^2, & if\ L \le K/a - K \\ [\lfloor K/a - L + 0.5 \rfloor I_i + \lfloor L + K - K/a + 0.5 \rfloor I_j]^2 + \\ [\lfloor K/a - L + 0.5 \rfloor Q_i + \lfloor L + K - K/a + 0.5 \rfloor Q_j]^2, & if\ L > K/a - K \end{cases}$$

Using (5) and (6), the statistical average is found as

$$W(L,K,a) = \begin{cases} K^2, & if\ L \le K/a - K \\ \lfloor K/a - L + 0.5 \rfloor^2 + \lfloor L + K - K/a + 0.5 \rfloor^2, & if\ L > K/a - K \end{cases} \tag{13}$$

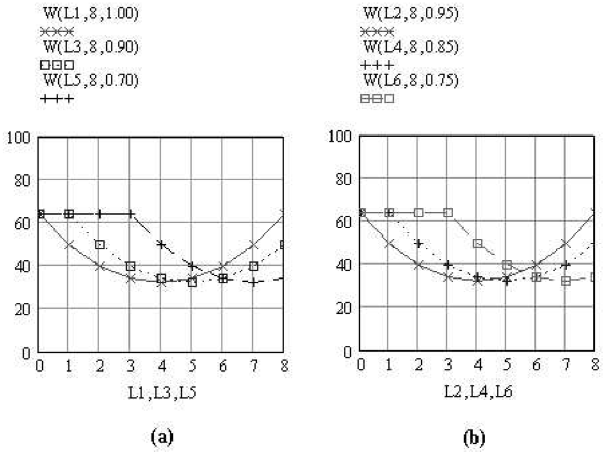
The plots are in Figure 2.10



(a) Case 5d: L=1, a=0.875

(b) Case 5d L=2, a=0.875

Figure 2.9: Case5d: the sampling frequency is higher than correct, and the window is delayed by LT$_l$. (a) the window covers the current symbol only. (b) the window covers the current symbol and part of the next symbol.



(a)

(b)

Figure 2.10: W(L, K, a) for Case 5d: (a) a=1,0.9,0.8, (b) a=0.95,0.85,0.75. Only points at integer L exist. The lines show the trend of change.

When the clock is advanced by LT$_l$ (case 5a, Figure 2.11), the window covers the previous symbol and the current symbol. The number of samples in the previous symbol is, and the number of samples in the current symbol is $M_I = K - L$. Thus

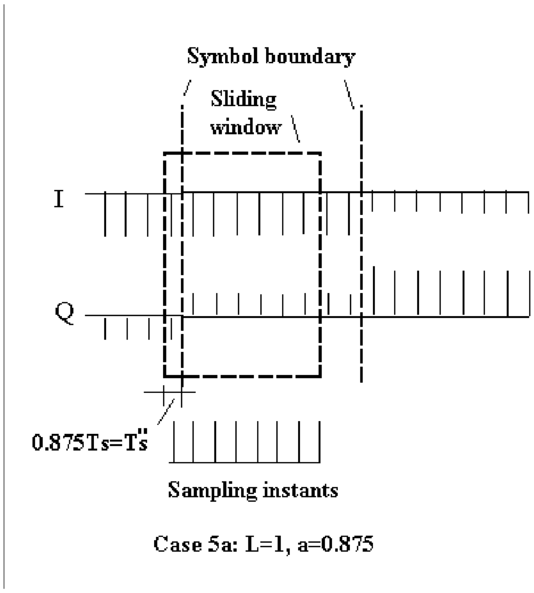


Figure 2.11: Case5a: the sampling frequency is higher than correct, and the window is advanced by LT$_l$. The window covers the current symbol and part of the previous symbol.

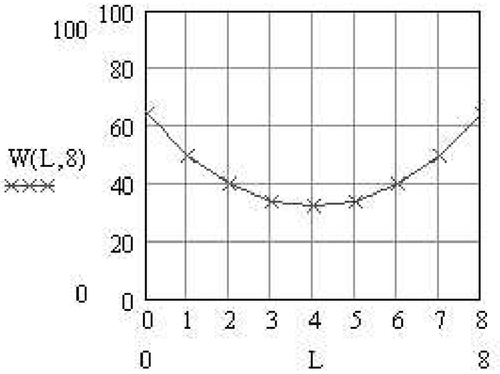


Figure 2.12: W(L,K) for Case 5a: independent of value of a. Only points at L equal to integers exist. The lines show the trend of change.

$$W(L,K,a,i,k) = [LI_k + (K-L)I_i]^2 + [LQ_k + (K-L)Q_i]^2$$

Using similar derivation as above, the statistical average is found as

$$W(L,K,a) = L^2 + (K-L)^2 \qquad (15)$$

The plots are in Figure 2.12.

2.4.3 Phase Detector Characteristic Summary

The average signal expressions are summarized in Table 2.1, where f_l and f_s are the local VCO frequency and received data symbol rate, respectively. The amplitude of the data is assumed as 1.

Table 2.1: W(L, K, a), L=delay or advance, K = number of correct clock period, $a = f_s / f_l$

Case	W(L, K, a)	Notes
1 synchronized, $f_l = f_s$	K^2	
2 delay L, $f_l = f_s$	$L^2 + (K-L)^2, \quad L < K$	
3 advance L, $f_l = f_s$	$L^2 + (K-L)^2, \quad L < K$	
4d delay L, $f_l < f_s$	$\left\lfloor \frac{K}{a} - L + 0.5 \right\rfloor^2 + \left\lfloor K - \frac{K}{a} + L + 0.5 \right\rfloor^2, \quad L \le (\frac{2}{a} - 1)K$	$a > 1$
4a advance L, $f_l < f_s$	$\begin{cases} L^2 + (K-L)^2, \; L \ge K(1-\frac{1}{a}) \\ L^2 + \left\lfloor \frac{K}{a} + 0.5 \right\rfloor^2 + \left\lfloor K - L - \frac{K}{a} + 0.5 \right\rfloor^2, \; L < K(1-\frac{1}{a}) \end{cases}$	$a > 1$
5d delay L, $f_l > f_s$	$\begin{cases} K^2, \; L \le \frac{K}{a} - K \\ \left\lfloor \frac{K}{a} - L + 0.5 \right\rfloor^2 + \left\lfloor L + K - \frac{K}{a} + 0.5 \right\rfloor^2, \; L > \frac{K}{a} - K \end{cases}$	$a < 1$
5a advance L, $f_l > f_s$	$L^2 + (K-L)^2, \quad L < K$	

2.5 Detailed Operation of the MSWS

The Modified Sliding Window Synchronizer is shown in Figure 2.2. The operation of the Modified Sliding Window Synchronizer is similar to the original SWS up to and including the creation of the W(n) analogue curve at the output of the S/P. The I and Q baseband signals are sampled at eight times the symbol rate and fed into shift registers with a length of eight taps. The samplers are controlled by a VCO or NCO. Figure 2.13 shows an example of the I and Q channels of the incoming symbol stream as well as the sampled points used by the MSWS. Figure 2.14 shows the contents of the shift registers for the first eight sample periods. It can be seen that the samples enter the shift register at tap one and shift the previous contents upward by one tap. Note the upward movement of the samples in the shift register in the plots from left to right. Every sample time, the contents of each shift register are summed and the resulting summations are squared as seen in Figure 2.15. The two resulting values are added and immediately fed into a S/P. The S/P has width of eight taps. The output of the S/P is referred to as the W(n) curve. The W(n) curve for the first six symbol periods can be observed in Figure 2.16.

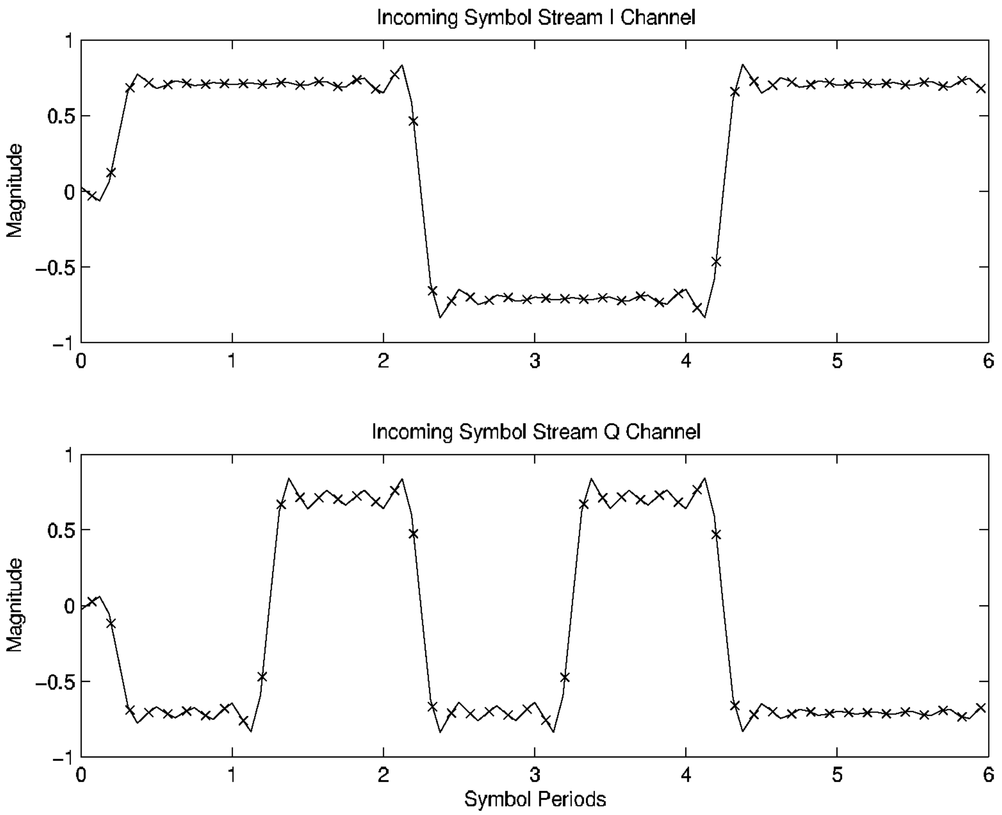


Figure 2.13: MSWS Incoming Symbol Stream Sampling

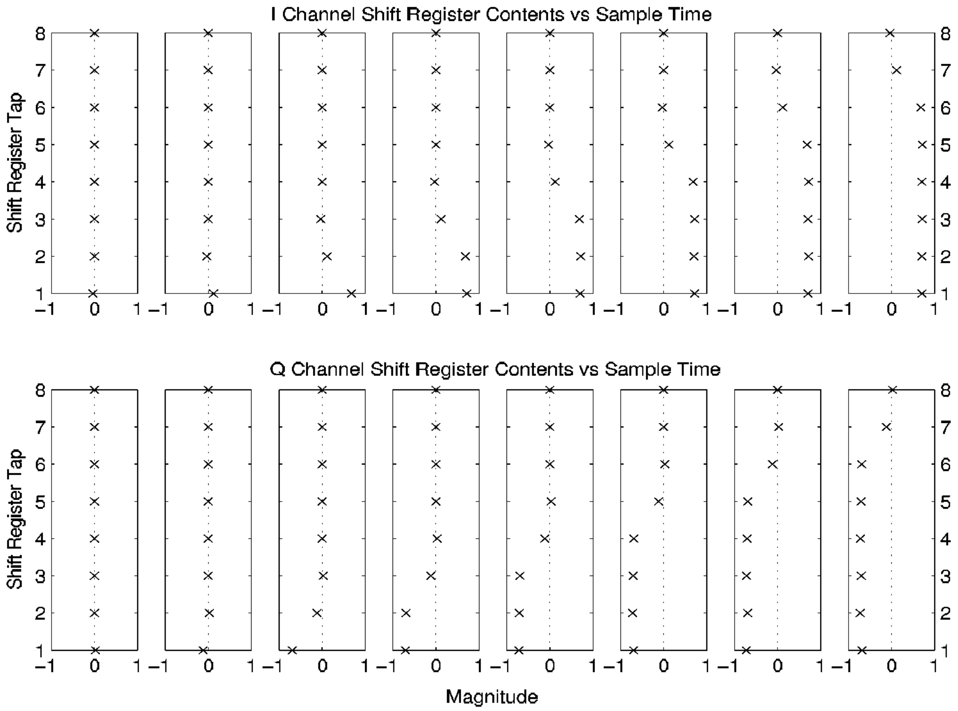


Figure 2.14: MSWS Shift Register Contents for First Incoming Symbol

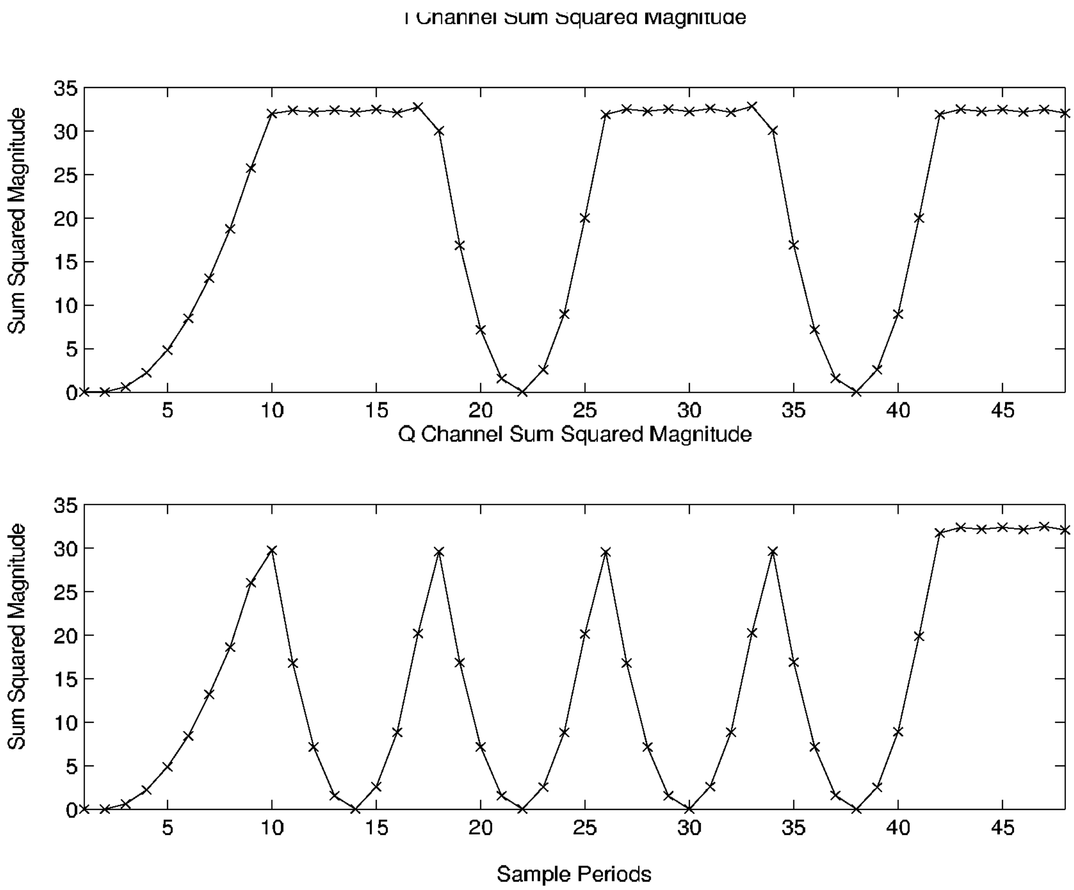


Figure 2.15: MSWS Sum of Squared Magnitude

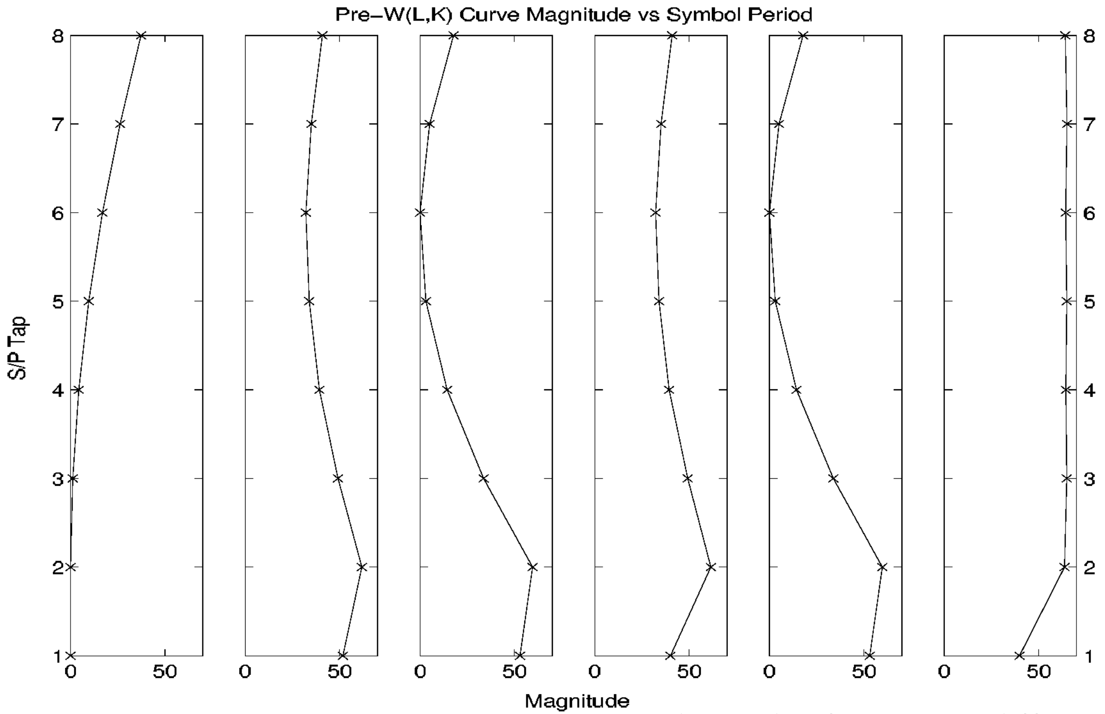


Figure 2.16: MSWS S/P Output vs. Symbol Period (Pre-W(L,K) curve or W(n)).

The S/P is followed by an SWA containing 32 taps. This SWA operates at the symbol rate. The SWA creates the long time average approximation of the *W(L,K)* curve. Figure 2.17 shows the development of the *W(L,K)* analogue curve at the output of the SWA over the first six symbol periods. The SWA is followed by a max tap selector, which selects the tap with the maximum magnitude. From the leftmost plot of Figure 2.17, it can be seen that tap 5 will be selected as the max tap. As a result, the output of the max tap selector will be the number five. The output of the max tap selector is used by the control algorithm in the creation of the control signal that is sent to the VCO or NCO. Additionally, it is used in the creation of the coarse adjustment signal. The control algorithm compares the output of the max tap selector to the previous output of the max tap selector and makes one of four decisions. These decisions are based on the difference between the current and previous outputs of the max tap selector. Table 2.2 shows the possible decisions and the resulting control signal generated. The control signal output is sent through a gain block and then to the VCO or NCO. The free running frequency of the VCO and NCO is equal to eight times the incoming symbol rate. The output of the VCO or NCO is sent to a clock divider to create the symbol-timing clock that controls the S/P. The symbol strobe output is a delayed version of the S/P control signal. The delay is determined by coarse adjustment signal created by the control algorithm. The operation of the Modified Sliding Window Synchronizer control algorithm is based on the same *W(L,K)* curve approximation as the original SWS. However, unlike the SWS technique, the length of the SWA is equal to 32 taps. The MSWS control algorithm operates on the

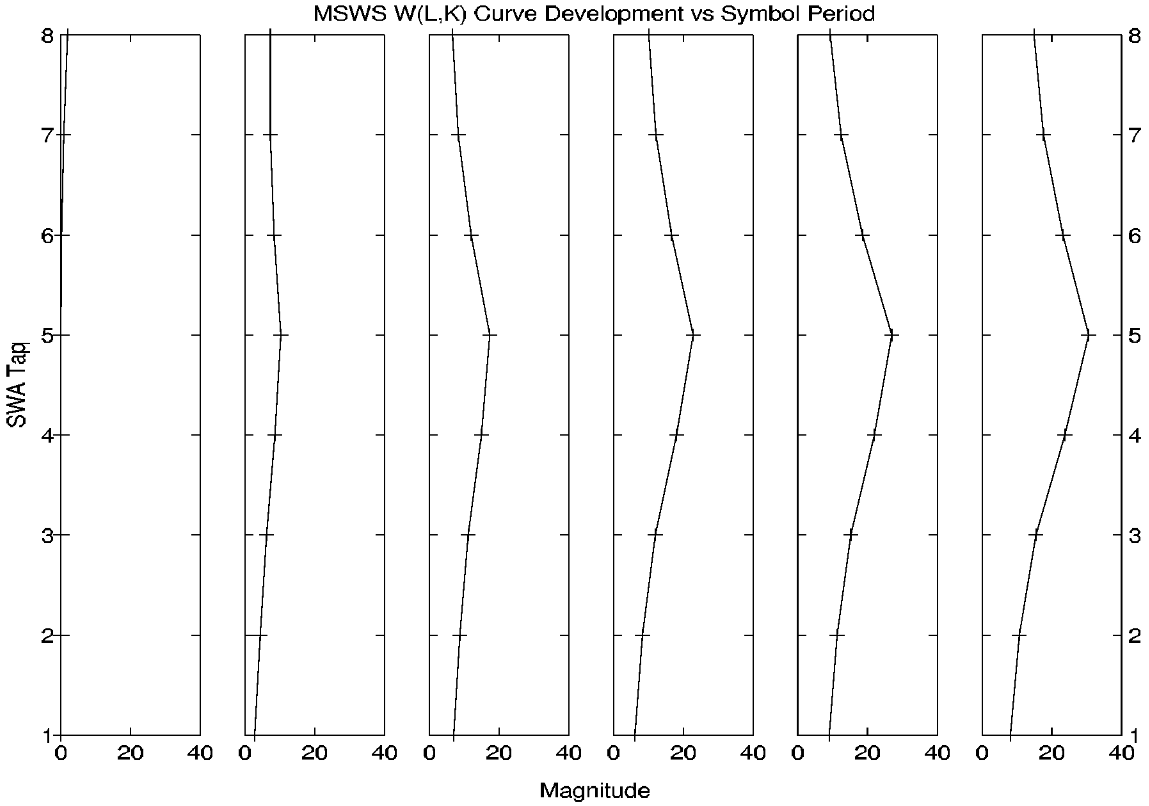


Figure 2.17: MSWS *W(L,K)* Analogue Curve Development

Current Max Tap Output Equals	Control Signal
MT[a]	SWA(NT)-SWA(PT)
NT[b]	SWA(NT)-SWA(PT)
PT[c]	SWA(NT)-SWA(PT)
other	Perform Coarse Adjustment

[a] MT is the previous output of the max tap selector
[b] NT is the tap immediately following MT
[c] PT is the tap immediately preceding MT

Table 2.2: MSWS Decision Table

whole *W*(*L*,*K*) analogue curve, as opposed to the two fixed taps involved in the SWS technique. From the entire curve, the tap with the greatest magnitude is selected and used to determine the fine adjustment signal as well as the coarse adjustment signal.

The control algorithm has two main modes of operation. The first mode is called coarse acquisition that attempts to quickly reduce the phase error to within one tap. This coarse acquisition is based on adjusting the delay of the symbol strobe with respect to the incoming baseband symbol stream in steps of whole sample times. The amount of delay is directly related to the index of the maximum tap. Since the *W*(*L*,*K*) curve can be approximated after only one symbol transition there is a greater than 98% likelihood that the output strobe can be brought within one tap within approximately four symbol periods. However, in the simulation model we perform the coarse acquisition every symbol period for the first ten symbols, which ensures that we will have a symbol transition in 999,999 out of 1,000,000 simulation runs. The nature of coarse acquisition also ensures that the algorithm will lock at the maximum point on the *W*(*L*,*K*) analogue curve. Figure 2.18 shows the incoming symbol stream and the symbol strobe signal for the first ten symbol periods. Note that the symbol strobe does not have fixed intervals in the region of the second and third symbol periods.

The second mode of operation is called fine adjustment and operates by adjusting the timing control output of the MSWS control algorithm. This, in turn, alters the output frequency of the VCO or NCO. The accumulated effects of the small frequency delta between the incoming symbol rate and the symbol-timing clock are used to drive the phase error to zero. When the frequency of the VCO or NCO is increased, the individual sample periods will be shorter and will have the cumulative effect of moving the symbol-timing clock earlier with respect to the incoming baseband symbol transitions. Decreasing the frequency of the VCO or NCO has the effect of lengthening the individual sample periods and has the cumulative effect of moving the symbol-timing clock later with respect to the incoming symbol transitions. Figure 2.19 shows the effect of adjusting the sampling rate. The symbol-timing clock used in the MSWS is the clock divided output of the VCO or NCO used to control the output of the S/P and SWA. Note that it does not necessarily have the same phase as the symbol strobe output. The disjoint between the symbol timing clock and the symbol strobe output allows for the operation of the coarse acquisition mode without destroying the contents the *W*(*L*,*K*) analogue curve, which would require that the SWA be flushed. The operation of the low-sample-rate version of the MSWS is similar to the high-sample-rate version. However, the low-sample-rate version operates with a nominal VCO or NCO

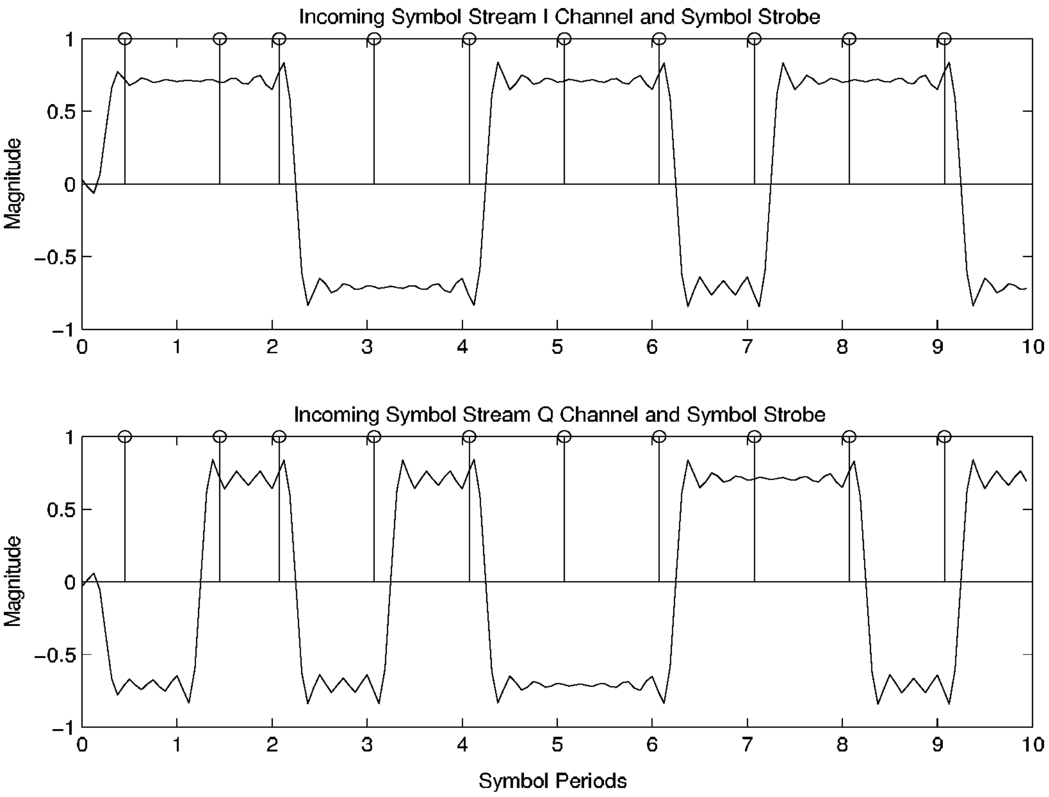


Figure 2.18: MSWS Symbol Strobe Output

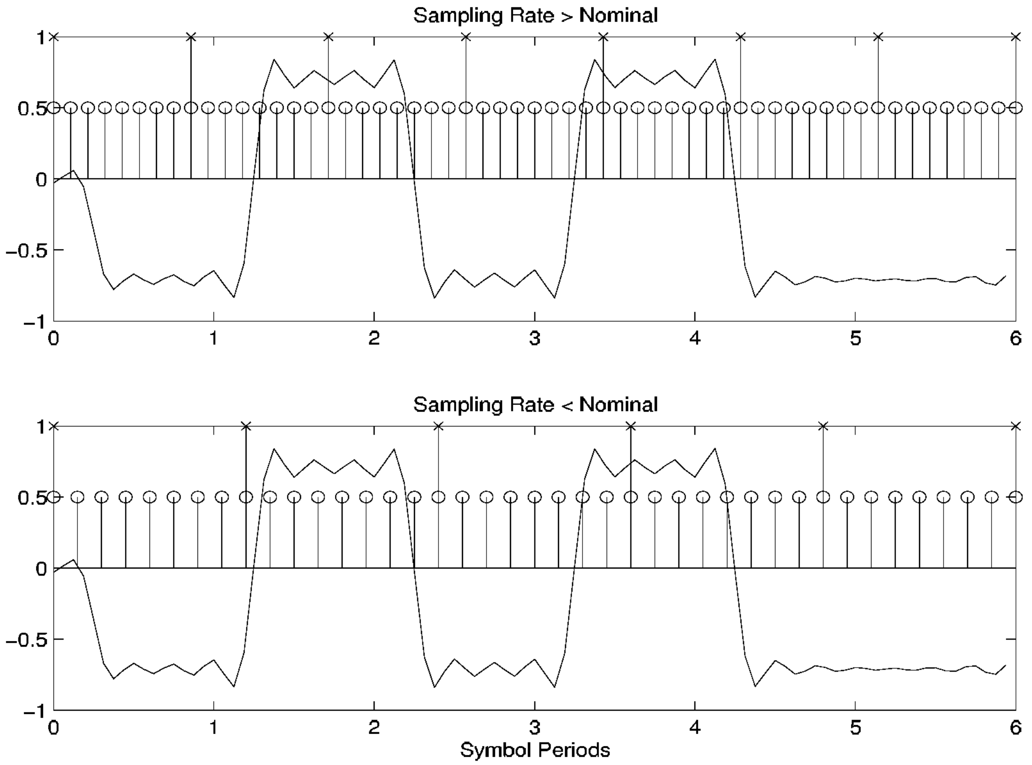


Figure 2.19: MSWS Sampling Rate Effect on Symbol Timing Clock

frequency of four samples per symbol. Additionally, the incoming shift registers and the S/P are four taps wide. The SWA following the S/P is four taps wide, but has the same length, 32 taps, as the high-sample-rate version. The operation of the max tap selector and control algorithm is identical to the high-sample-rate version. The operation of the coarse acquisition and fine adjustment are the same in the low and high-sample-rate versions.

2.6 Observation

Like the original Sliding Window Synchronizer algorithm, the fine adjustment technique of the Modified Sliding Window Synchronizer asymptotically oscillates the phase error to zero (see next Section). However, the use of the coarse acquisition technique allows the MSWS to quickly achieve a phase error of less than one tap. In the high-sample-rate version of the MSWS this equals a phase error that is less than 0.125 symbol periods. The phase error for the low-sample-rate modification is less than 0.25 symbol periods. When used with digital demodulators and square symbol pulses, the MSWS can achieve the theoretical performance of the demodulator within about four symbol periods. Non-square symbol pulse operation or other synchronization applications requiring less phase error will necessarily wait for the fine adjustment to achieve the required phase error.

3. Synchronizers For Comparison

3.1 The Early-Late Gate Synchronizer

The Early-Late Gate Synchronizer compares the signal energies in an early gate and a late gate (Figure 3.1). If the symbol timing is correct, the difference is zero; otherwise, the difference is positive or negative. The polarity and the magnitude of the difference signal indicated that whether the timing clock is earlier or late and how much is advanced or delayed, with respect to the received data pulses. This signal is used to control the VCO.

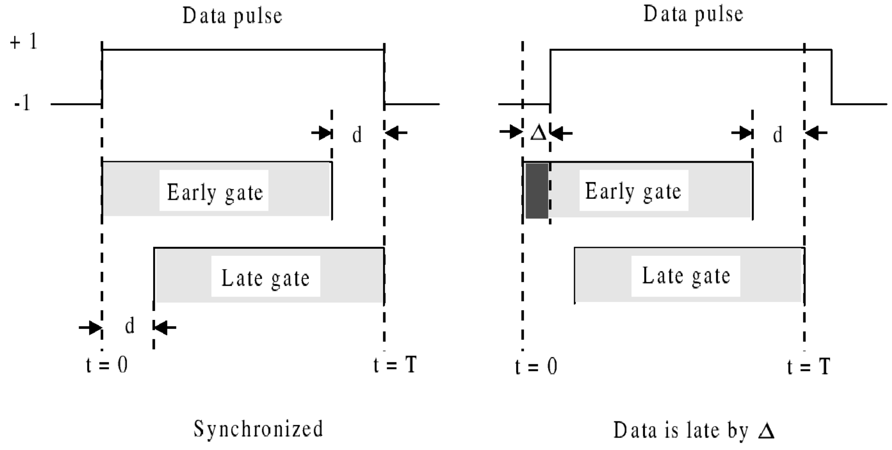


Figure 3.1. Early-Late Gate Timing Illustration

The Early-Late Gate Synchronizer is a modification of the traditional analog ELGS [7], which operates on samples of the baseband symbol stream. The ELGS is presented as a baseline technique. Its operation is rather simple and it is easily realizable in hardware. Like the SWS and MSWS both high and low-sample-rate models are possible. The Early-Late Gate Synchronizer is shown in Figure 3.2. The ELGS operates on either the I or Q channel of the baseband signal. The incoming baseband signal is sampled at eight times the incoming symbol rate.

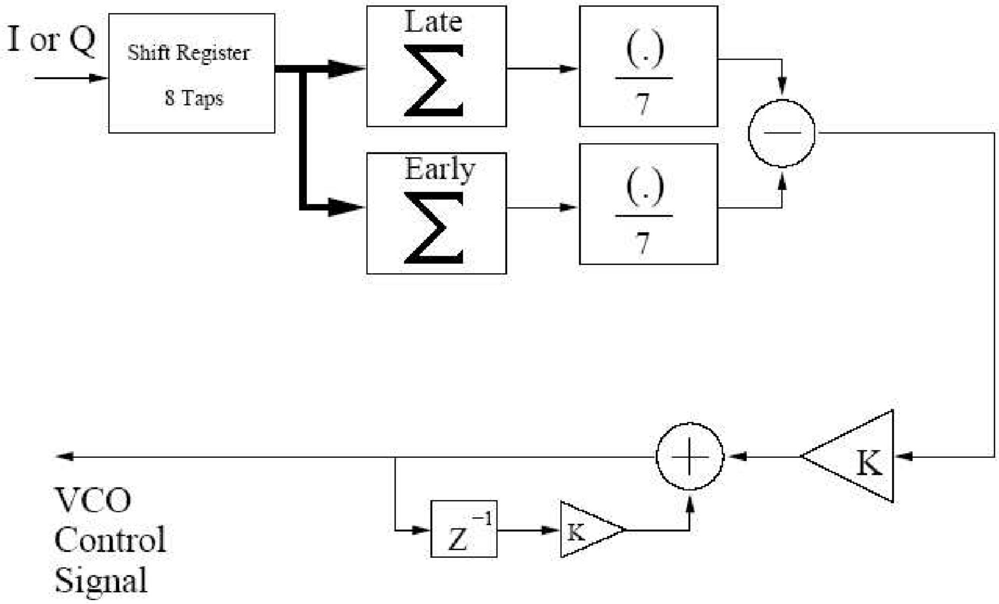


Figure 3.2: Early-Late Gate Synchronizer Block Diagram

These samples are then sent to a shift register of length eight taps. Samples enter at tap one and shift the previous contents further into the shift register. Once each symbol period, the contents of the shift register are summed twice. The first summation, called the early summation, sums all but the last tap of the shift register. The second summation, called the late summation, sums all but the first tap of the shift register. Each of these two summations is divided by the number of taps included in each summation. In this way, the early and late integrators of the analog ELGS are approximated in the digital ELGS.

The operation of the ELGS is best understood by an example. Figure 3.3 shows the samples chosen for the early and late summations. In the plot, the circle-topped lines indicate the early summation samples and the square topped lines indicate the late summation samples. Figure 3.4 shows a plot of the output of the early and late summations over the first six symbol periods. Comparing this to Figure 3.3, note that the summations are equal every time there is a consecutive symbol of the same phase. The early summation is subtracted from the late summation and the result is sent to the digital integrator and loop filter. The output of the loop filter is used to control the output of the VCO. The VCO is set to a free-running frequency of eight times the incoming symbol rate. The output of the VCO is the intermediate timing signal. This is used to control the samplers and is also sent to a clock divider to create the symbol strobe signal. The symbol strobe signal is used to control the summation of the delay lines and has the same frequency and phase as the incoming symbol data. Figure 3.5 shows the incoming baseband symbol stream and the symbol strobe created by the ELGS for the first six symbol periods. It should be obvious from the plot that the ELGS is unable to acquire phase lock over this small period of time. The low-sample-rate version of the Early-Late Gate Synchronizer sets the free-running frequency of the VCO to four times the symbol frequency. It also decreases the length of the incoming delay line to four taps.

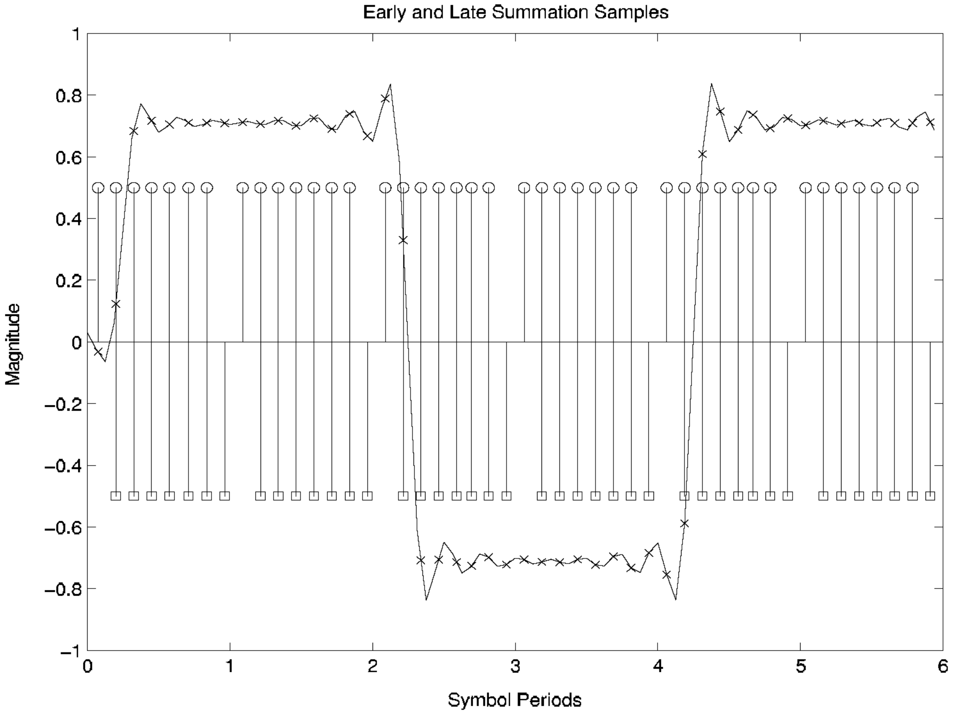


Figure 3.3: ELGS Early and Late Summation Sampling Points

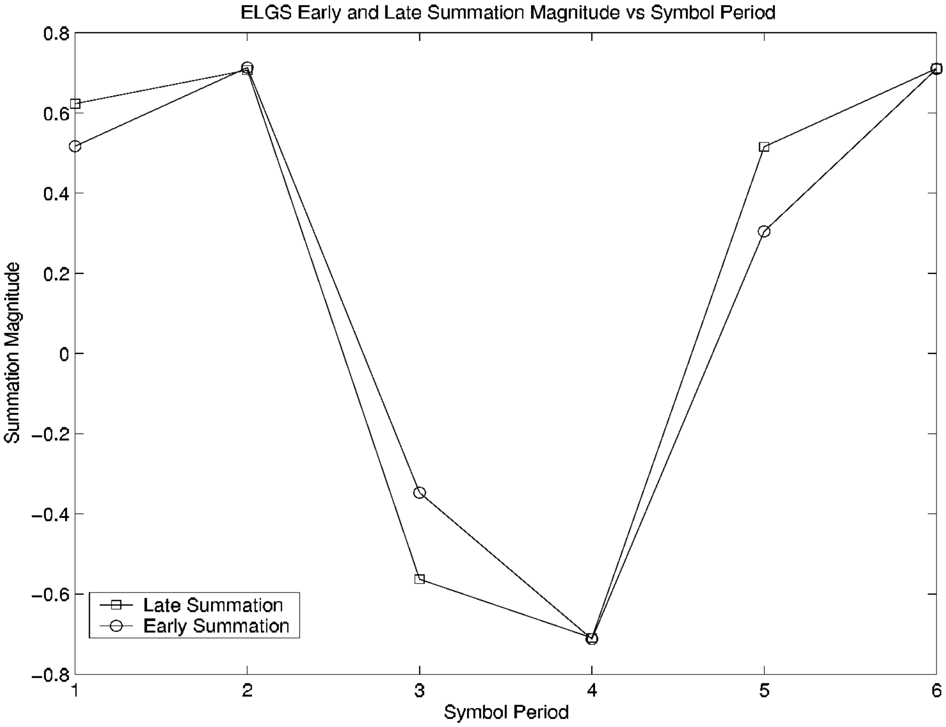


Figure 3.4: ELGS Summation Magnitude vs. Symbol Period

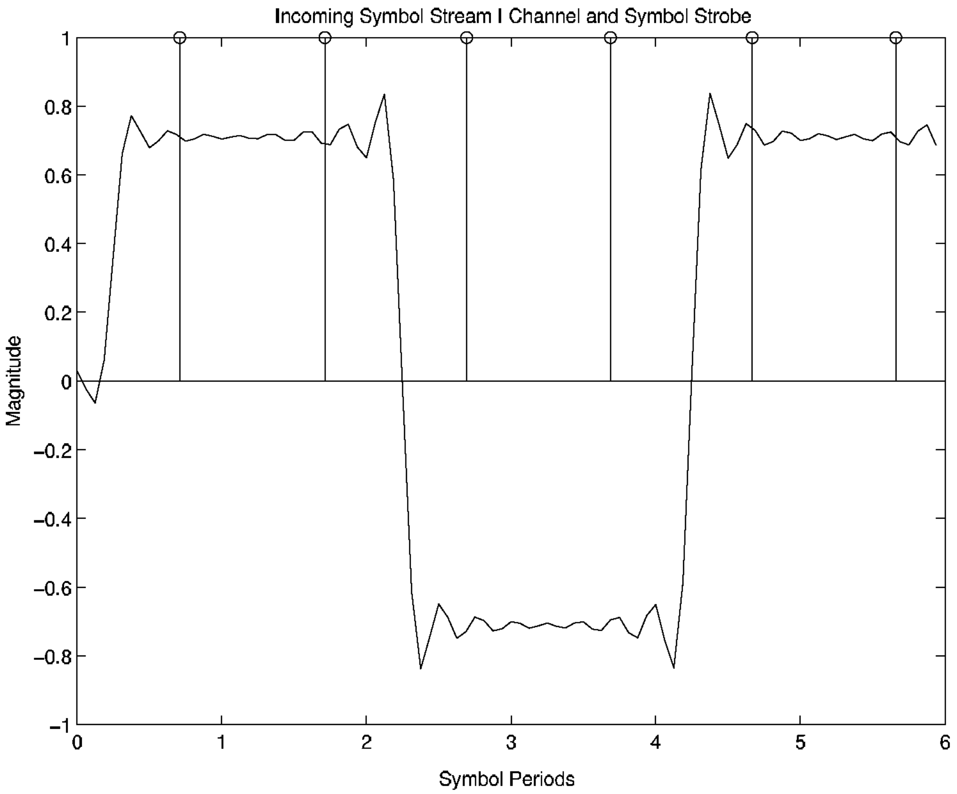


Figure 3.5: ELGS Symbol Strobe Output

This means that the early summation will be over the first three taps and the late summations will be over the last three taps. Additionally, the summations will be divided by three instead of seven, as is the case in the high-sample-rate version.

3.2 Gardner Zero-Crossing Detector

The Gardner Zero-Crossing Detector was first proposed by F.M. Gardner as a simple timing error detector for BPSK and QPSK communications systems [4]. This scheme is an ad hoc technique that operates at two samples per symbol. There have been several proposed methods of improving the performance of the GZCD. This work, however, only investigates the original algorithm.

The Gardner Zero-Crossing Detector operation is based on the following symbol timing error characteristic

$$u_t(r) = y_I(r-1/2)\{y_I(r) - y_I(r-1)\} + y_Q(r-1/2)\{y_Q(r) - y_Q(r-1)\} \tag{16}$$

which can be expressed as the subtraction of the current symbol strobe samples from the previous symbol strobe samples and the multiplication of the results by the sample occurring midway between the current and previous strobe samples and finally by the addition of the two I- and Q-channel results. Derivation of this expression is later given in Section 3.3. Since the GZCD samples at two times the symbol rate, the samples occurring midway between the symbol strobe samples should lie in the transition region if there is a transition between symbols. When timing error is zero, the midway sample should give an average value of zero. Timing error results in a nonzero sample whose magnitude depends on the amount of error. The midway sample does not provide enough information alone to control the synchronizer, as either slope is equally likely. In order to eliminate the ambiguity, the symbols strobe samples preceding and following the midway sample are examined. If there is no symbol transition, the strobe values will be the same, and their difference zero, so the midway sample is rejected. If there is a symbol transition, the strobe values will be different and the difference will provide the slope information and the midway sample will provide the timing error information.

The Gardner Zero-Crossing Detector is shown in Figure 3.6. The incoming baseband signal is sampled in both the I and Q channels. The VCO, which controls the samplers, operates at two times the symbol rate. A plot of the incoming baseband signals and the sample points used by the GZCD is shown in Figure 3.7. The output of each of the samplers is sent to a shift register of length three taps. The shift registers are sampled once per symbol period by a sample and hold device (S/H). The output of the S/H contains the strobe samples of the current and previous symbol as well as the sample that lies midway between them. These are located at taps one, three, and two, respectively. For each channel, the previous strobe sample is subtracted from the current strobe sample. The result of this subtraction is then multiplied by the midway sample. The result from the I branch and the Q branch are added together and sent to the loop filter and integrator. The output of the loop filter and integrator is used to control the VCO. The output of VCO is sent to a clock divider that reduces the frequency from the sampling rate to the symbol rate. The output of this clock divider is the symbol strobe signal and is used to control the S/H. Figure 3.8 shows the incoming symbol stream and the symbol strobe created by the GZCD.

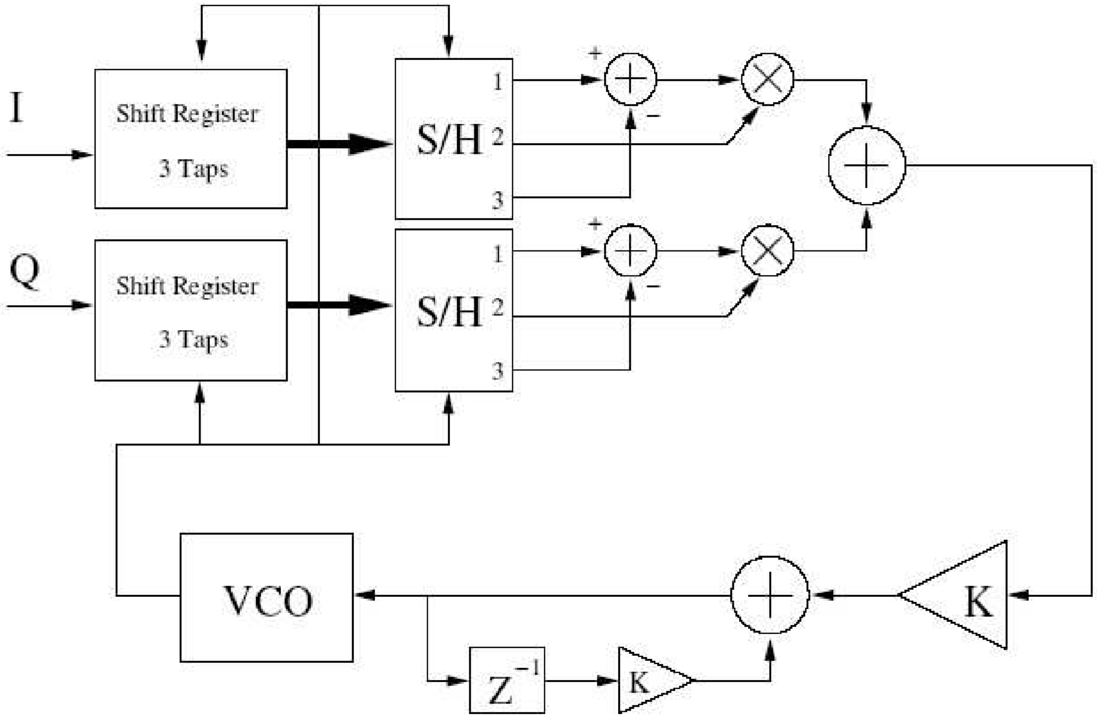


Figure 3.6: Gardner Zero-Crossing Detector Block Diagram

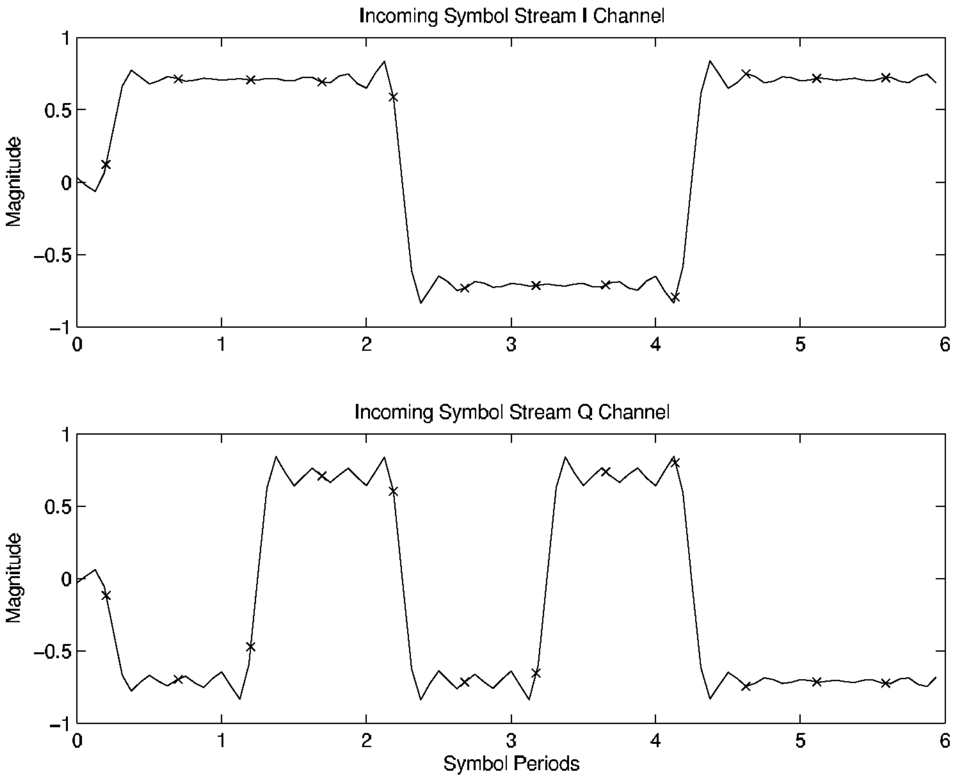


Figure 3.7: Gardner Zero-Crossing Detector Incoming Symbol Stream Sampling

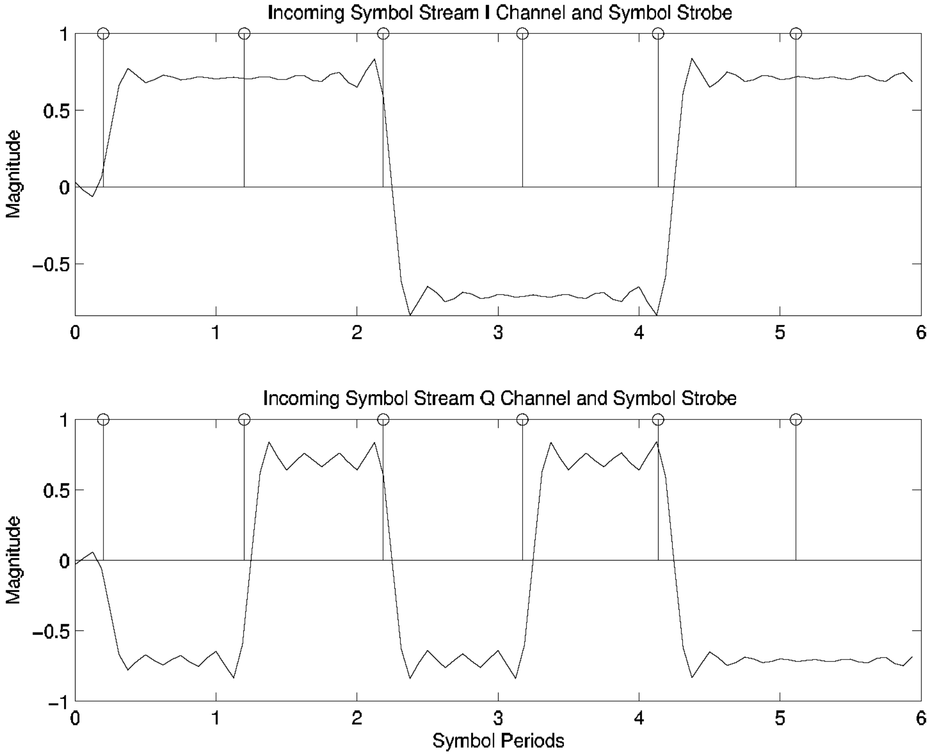


Figure 3.8: Gardner Zero-Crossing Detector Symbol Strobe Output

3.3 Derivation of symbol timing error characteristic of the GZCD (Eqn.16):

The Gardner Zero-Crossing Detector's operation begins with a pair of baseband samples, $y_I(t)$ and $y_Q(t)$ from the I and Q channels. The GZCD operates on two samples per symbol period, T. As a result, the ideal samples of the rth symbol are expressed as follows.

$$y_I(rT),\ y_Q(rT),\ y_I((r-1/2)T),\ y_Q((r-1/2)T)$$

where rT is the sampling time occurring at the rth symbol's strobe time and (r-1/2)T is the sampling time lying midway between the rth and (r − 1)th symbols' strobe times. For practical systems, the synchronizer's sampling points will delayed by time τ, resulting in the following samples for the rth symbol.

$$y_I(rT+\tau),\ y_Q(rT+\tau),\ y_I((r-1/2)T+\tau),\ y_Q((r-1/2)T+\tau)$$

The delay difference signal, $x_d(t)$, can be used to approximate a differentiator, which is often used in analog clock recovery techniques.

$$x_d(t) = x(t) - x(t-t_d)$$

If t_d is chosen to be T/2, it can be seen that the GZCD samples from either the I or Q channel can be used to create the delay difference signal.

Squaring the delay difference leads to the following equation.

$$x_d^2(t) = x^2(t) + x^2(t-T/2) - 2x(t)x(t-T/2)$$

Sampling at $t = rT+\tau$ and $t = (r-1/2)T+\tau$ gives two additional equations.

$$E(r) = x_d^2(rT+\tau)$$

$$L(r-1) = x_d^2((r-1/2)T+\tau)$$

Note that $L(r-1)$ utilizes the sample midway between the (r −1)th and rth symbol strobe samples. These equations are expanded as follows.

$$E(r) = x^2(\tau+rT) + x^2(\tau+(r-1/2)T) - 2x(\tau+rT)x(\tau+(r-1/2)T)$$

$$L(r-1) = x^2(\tau+(r-1/2)T) + x^2(\tau+(r-1)T) - 2x(\tau+(r-1/2)T)x(\tau+(r-1)T)$$

The GZCD detector algorithm is defined as the following equation.

$$u_t(r) = L(r-1) - E(r)$$

Expanded and simplified, this leads to the following equation.

$$u_t(r) = x^2(\tau+(r-1)T) - x^2(\tau+rT) + 2x(\tau+(r-1/2)T)\{x(\tau+rT) - x(\tau+(r-1)T)\}$$

The GZCD relies on the average over many samples of the detector algorithm for its operation. This allows for the elimination of the x^2 terms as they do not contribute to the average output. The only output that they could produce would be due to noise, as they would cancel out completely in the absence of noise. This leads to the following detector algorithm.

$$u_t(r) = x(\tau+(r-1/2)T)\{x(\tau+rT) - x(\tau+(r-1)T)\}$$

Expressed more compactly, it is.

$$u_t(r) = x(r-1/2)\{x(r) - x(r-1)\}$$

The final detector algorithm consists of adding the results of the detector algorithm from the I and Q channels. The GZCD generates one error sample $u_t(r)$ for each symbol.

$$u_t(r) = y_I(r-1/2)\{y_I(r) - y_I(r-1)\} + y_Q(r-1/2)\{y_Q(r) - y_Q(r-1)\}$$

which is (16). For further information on the Gardner Zero-Crossing Detector please refer to [4].

4. Synchronizers Performance Evaluation

The robustness of each of the four synchronization schemes discussed in this work is evaluated by quantitative and qualitative methods. Methods for comparing these schemes are outlined in Section 4.1. A description of the methods of performance evaluation is provided in Section 4.1. A detailed examination of each synchronization scheme follows in Sections 4.3, 4.4, 4.5 and 4.6

4.1 Performance Metrics

This section proposes and defines several performance metrics by which the robustness of a synchronization scheme may be quantitatively determined. These metrics are derived from actual engineering application requirements. The metrics are convergence time, mean-square phase error, lowest SNR for locking, initial frequency offset performance, midstream frequency offset performance, and system complexity. Additionally, other pertinent advantages and disadvantages of each synchronizer will be examined.

Convergence time is determined by the average number of symbols required for each synchronization scheme to reach a predetermined phase error. This value determines how long the communications receiver will be operating at an elevated symbol error rate (SER) when the system is first started and again in the event that resynchronization is required. In modern high bit-rate communication systems the differences in this metric can determine whether higher-level protocols such as TCP/IP will suffer dramatic retransmission penalties in the event that resynchronization is required.

Mean-square phase error is determined by the difference between incoming symbol edges and the synchronizer's strobe signal when the synchronizer has reached the steady state. It is calculated by taking the average of the square of the difference between the synchronizer's estimated symbol timing and the correct symbol timing, where the difference is given in terms of fractions of a symbol period. In a digital communications system, a synchronization phase error results in the incorrect sampling of the incoming symbol. This, in turn, results in ISI and as a result produces symbol errors even in the absence of noise [5] . Phase error results in a error floor in the SER vs. SNR graph of the communications system. Figure 4.1 shows the effect of phase error on the SER performance of a QPSK communication system in AWGN [5]. BPSK-based communications systems show a similar error floor.

The lowest SNR for locking is determined by operating each synchronizer over a range of noise levels. Starting from the highest SNR, the noise level is increased until the synchronizer is unable to obtain lock. The SNR associated with this noise level is then called the lowest SNR for locking. All SNR values are given in terms of E_b/N_o. The lowest SNR for locking determines whether a synchronization scheme is suitable for a communication system with a given transmitter power and channel noise. This metric also determines the synchronizer's immunity to bursts of noise.

Initial frequency offset performance indicates a synchronizer's capacity to handle a frequency difference between its own clock and the transmitter's clock at the beginning of a transmission. The typical frequency offset in communications systems is usually on the order of 100 ppm . The nominal sampling rate of each synchronizer is varied over a range of frequency offsets, and the ability to obtain and to maintain lock is determined. Furthermore the mean-square phase error of the simulations is a measure of the synchronizer's frequency offset performance after the synchronizer has reached the steady state.

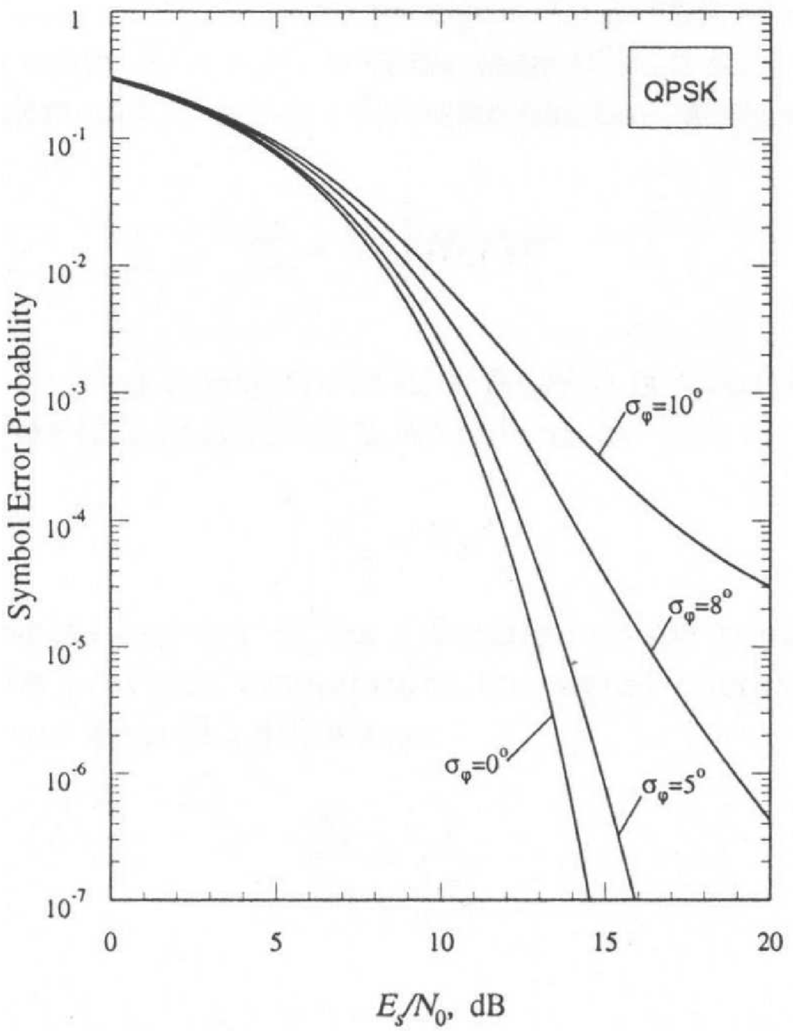


Figure 4.1: Noise Floor Effect Due to Incorrect Sampling

Midstream frequency offset performance of a synchronizer indicates how well it adapts to a frequency offset between the synchronizer and transmitter clocks which may occur in the middle of a transmission. Frequency offsets on the order of 100 ppm are typical in modern communications systems utilizing crystal oscillators. The midstream frequency offsets are performed after the synchronizers have achieved lock. The ability of the synchronizer to maintain lock despite the offset will be measured by observation of the phase offset plot in the region of interest.

The system complexity of a synchronizer is determined by two main elements, operating sampling rate and algorithm complexity. The combination of sampling rate and algorithm complexity determines the total computational requirement of the synchronization scheme. A synchronizer that requires a higher sampling rate will often require a more expensive ADC. Additionally, synchronizers with greater algorithm complexity will often require faster DSP.

4.2 Methods of Performance Evaluation

The performance evaluation of the robust synchronization candidates is performed using Matlab simulations. Each of the synchronizers is modeled as a separate Matlab program. Additionally, each synchronizer requires an incoming baseband symbol stream. Matlab based simulation platform provides the baseband symbol stream. The simulation platform performs the operations of a QPSK modulator, a band-limited AWGN channel, and a demodulator front-end. Figure 4.2 shows the block diagram of the simulation platform. Note that all of the synchronizers operate on one or both of the I and Q baseband signals produced by the simulation platform.

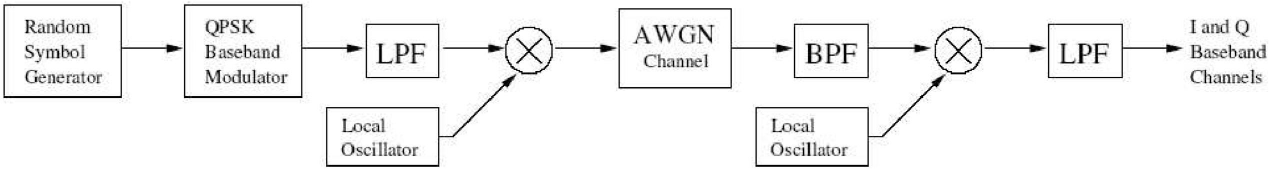


Figure 4.2: Simulation Platform Block Diagram

Several small Matlab scripts (Appendix A) are used to control the simulation of the synchronizers in order to evaluate their performance. The offset control script iterates through several initial phase offsets and captures the phase error signal from the synchronizer model being studied. The offset control script keeps a constant SNR over the entire range of phase offsets. The resulting phase error signals are used to determine the synchronizer's initial phase offset locking range as well as its convergence time. The SNR control script iterates through a range of SNR and captures the phase error signal from the synchronizer model being studied. The mean-square phase error at each SNR is calculated from the resulting phase error signals. The SNR control script also contains a larger loop that performs the SNR iterations several times so that an average mean-square phase error can be calculated. The frequency-offset script iterates through a range of frequency offsets and captures the phase error signal from the synchronizer. The phase error signals are then used in the initial and midstream frequency offset performance evaluation.

4.3 Sliding Window Synchronizer Performance

4.3.1 Convergence Time

The SWS convergence time is fairly long. Many symbols are required in order for the incremental phase differences between the incoming symbol stream and the symbol strobe to be reduced to less than a one tap phase error. The high-sample-rate SWS can take up to 350 symbol periods to achieve less-than-one-tap phase error. Convergence plots of the high-sample-rate SWS are shown in Figure 4.3. The offsets range from zero to one symbol periods in steps of 0.2 symbol periods. All of the convergence plots in Figure 4.3 are performed at an SNR (E_b/N_o) of 10 dB. The worst case offset, one half of the symbol period, for the low-sample-rate SWS requires over 400 symbol periods to achieve less than one tap phase error at a SNR of 10 dB. A plot of the convergence time of the low-sample-rate SWS is shown in Figure 4.4. The initial offsets are from zero to one symbol periods in steps of 0.2 symbol periods.

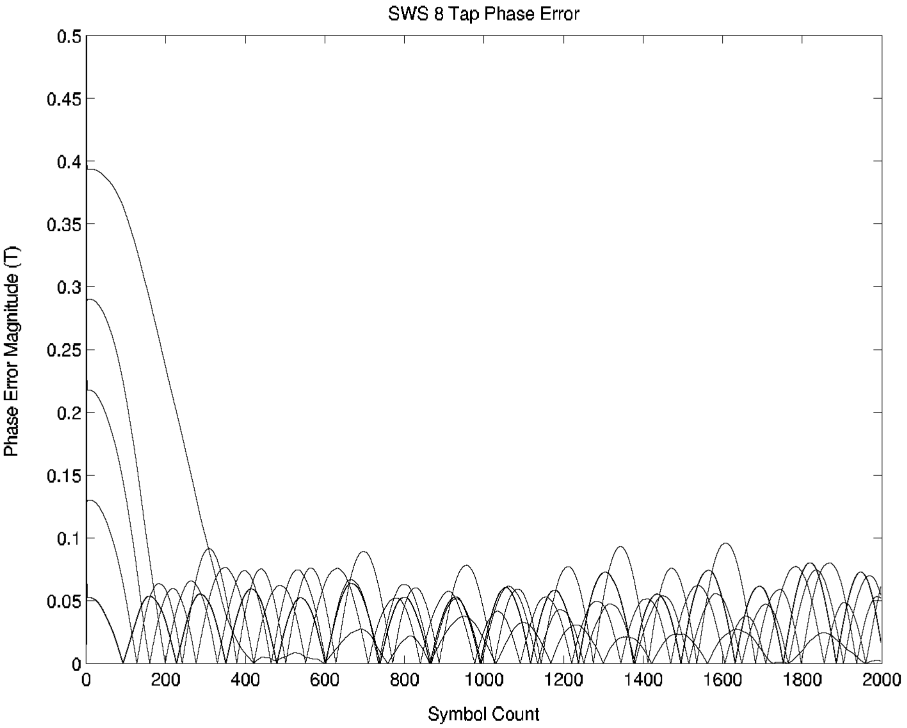


Figure 4.3: High-sample-rate SWS Convergence Plot

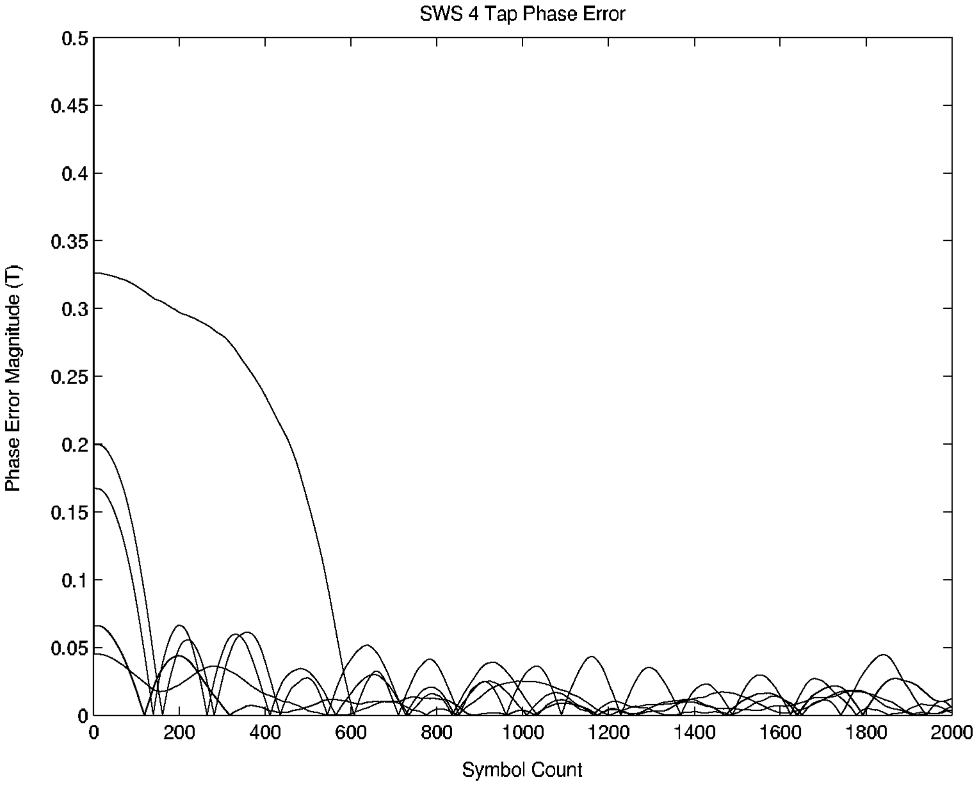


Figure 4.4: Low-sample-rate SWS Convergence Plot

There is no convergence to zero phase error in either the high or the low-sample-rate versions of the SWS. This may be problematic for systems that use non-square pulse shapes. It may also introduce some ISI. It seems that the convergence of the SWS in single-carrier systems is rather poor.

4.3.2 Mean-Square Phase Error

The mean-square phase error performance of the SWS was not expected to be extraordinary. Since the SWS was designed for OFDM systems, it is not surprising that its performance in single-carrier is lacking. In our evaluation, the high-sample-rate version posts a lowest mean-square phase error of 6.7×10^{-4} T^2 at 10 dB and a worst-case mean-square phase error of 1.4×10^{-3} T^2 at 3 dB. A plot of mean-square phase error over a range of SNR from 0 dB to 10 dB is shown in Figure 4.5. The low-sample-rate version of the SWS performs better than the high-sample-rate version. AN explanation for this phenomenon is that the low and high-sample-rate versions both use the same length SWA. This gives the low-sample-rate version an advantage in terms of the number of symbols over which its phase detector characteristic operates compared to the high-sample-rate version. Its lowest mean-square phase error is 1.3×10^{-4} T^2 at 10 dB, but its worst-case mean-square phase error is 5.2×10^{-4} T^2 at 3 dB. Figure 4.6 shows the mean-square phase error for the low-sample-rate SWS over a range of SNR from 0 dB to 10 dB.

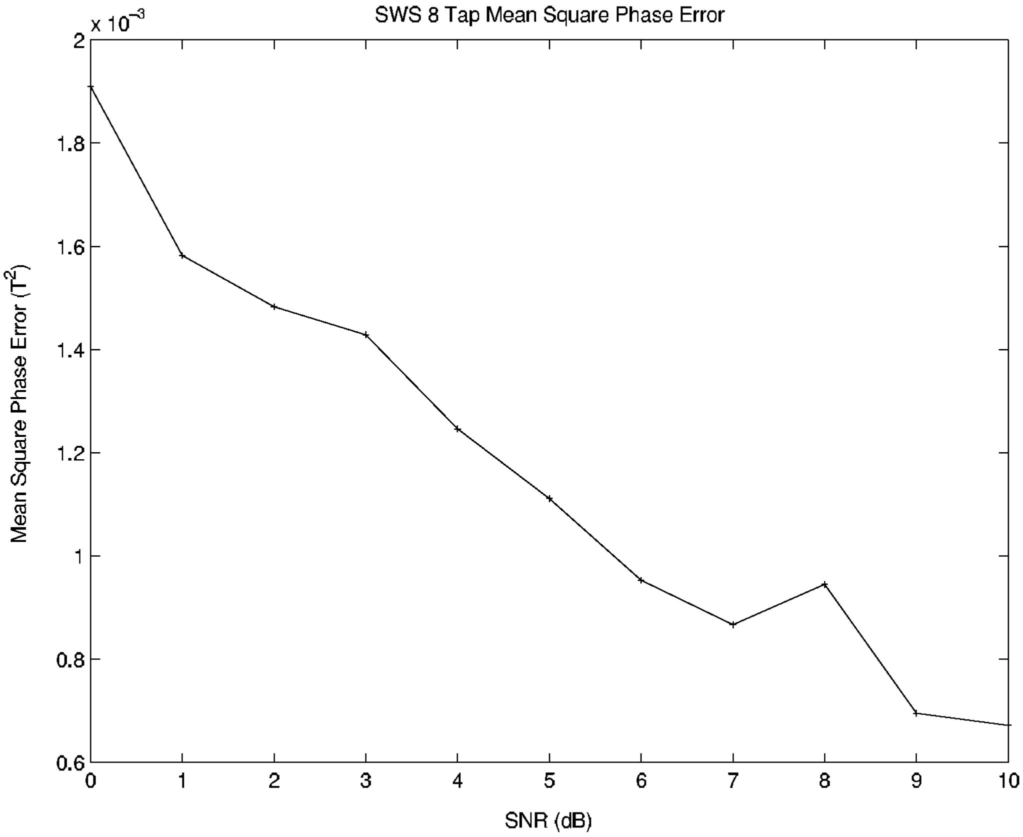


Figure 4.5: High-Sample-Rate SWS Mean-Square Phase Error Plot

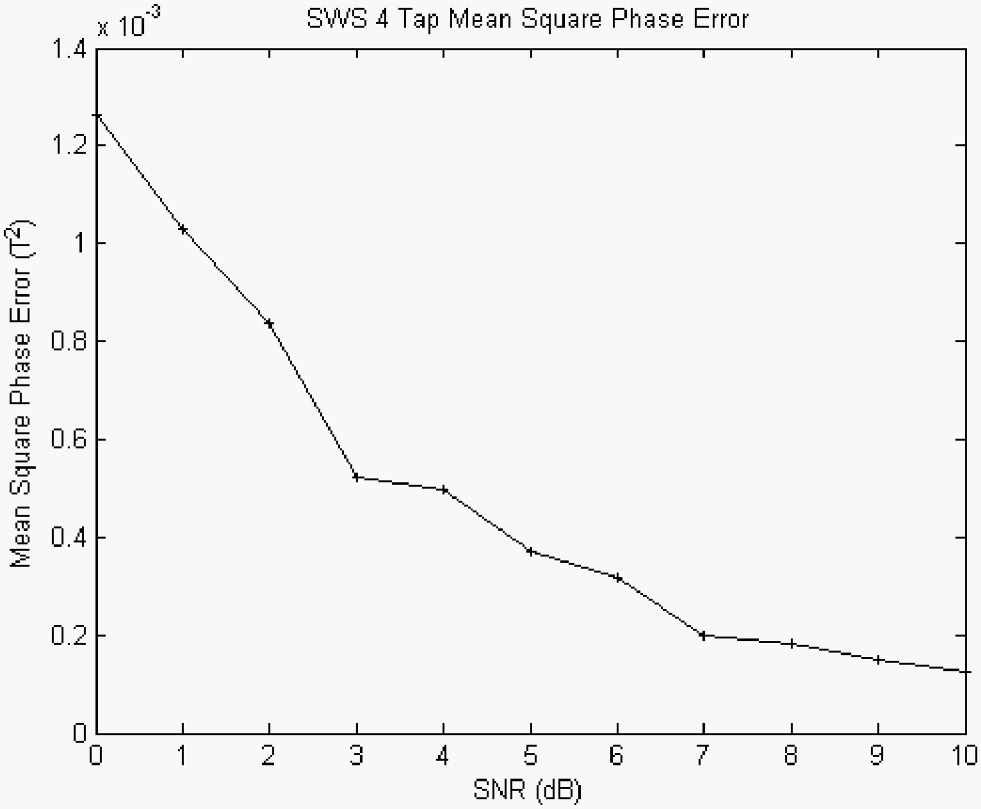


Figure 4.6: Low-Sample-Rate SWS Mean-Square Phase Error Plot

4.3.3 Lowest SNR for Locking

The SWS was expected to perform at SNR (E_b/N_o) as low as 0 dB as published results have previously shown [3]. However, those expectations are in OFDM systems. The actual simulation results show a lowest SNR for locking to be 3 dB in single-carrier systems. A problem that plagues the SWS in single-carrier systems is its inability to converge to zero phase error. This can clearly be seen for the high-sample-rate SWS in Figure 4.7. The initial offset for these plots is 0.3 symbol periods.

The low-sample-rate version of the SWS also is unable to achieve zero phase error. It seems to have a more prominent threshold at its lowest SNR for locking than does the high-sample-rate version. In Figure 4.8, note that the low-sample-rate SWS performs in a typical fashion at a SNR of 3 dB. However, at a SNR of 2 dB the low-sample-rate SWS is unable to operate correctly. The offset for these plots is 0.3 symbol periods.

4.3.4 Frequency Offset Performance

SWS is not able to perform over the entire range of frequency offsets from -100 to +100 ppm . The ability to maintain synchronization lock is lost at frequency offsets greater in magnitude than 50 ppm . Figure 4.9 shows the loss of synchronization lock for the high-sample-rate SWS. This plot is performed with an offset of 0.3 symbol periods and at a SNR of 10 dB. The mean-square phase error performance of the SWS over the same range of offsets is shown in Figure 4.10.

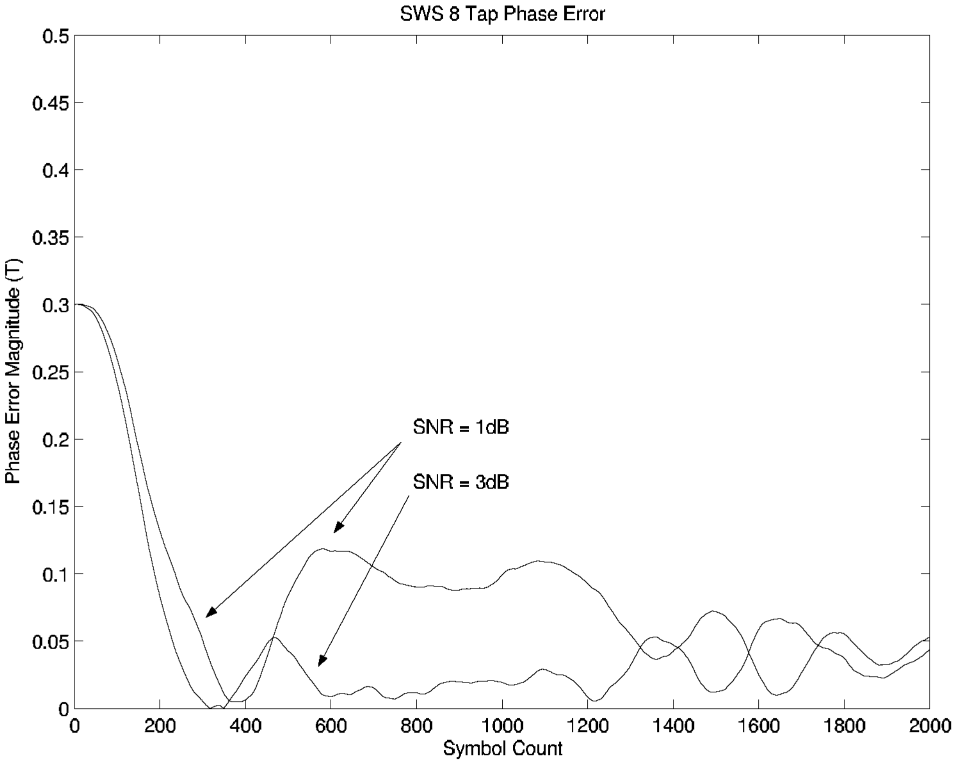


Figure 4.7: High-Sample-Rate SWS Low SNR Plot

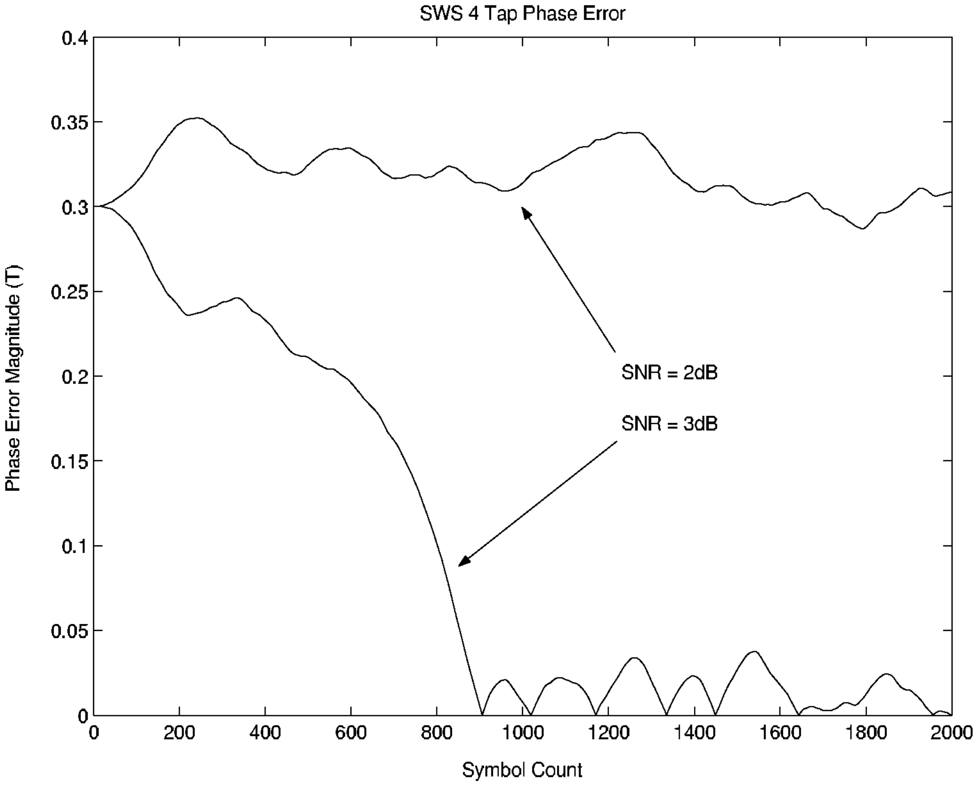


Figure 4.8: Low-Sample-Rate SWS Low SNR Plot

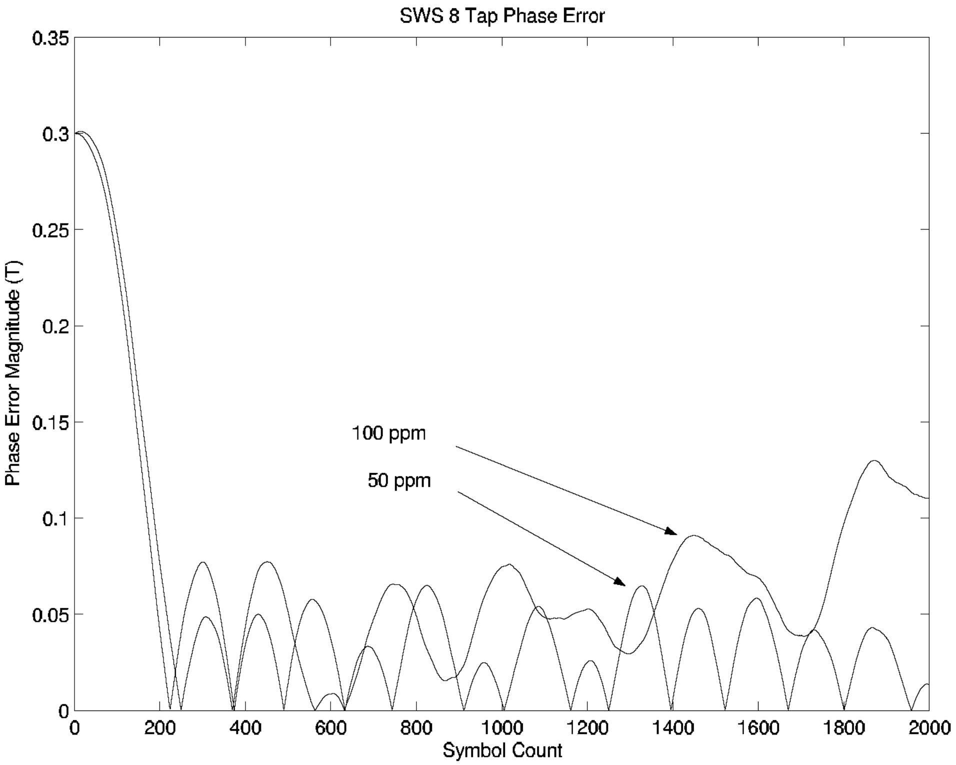


Figure 4.9: High-sample-rate SWS Frequency Offset Convergence Plot

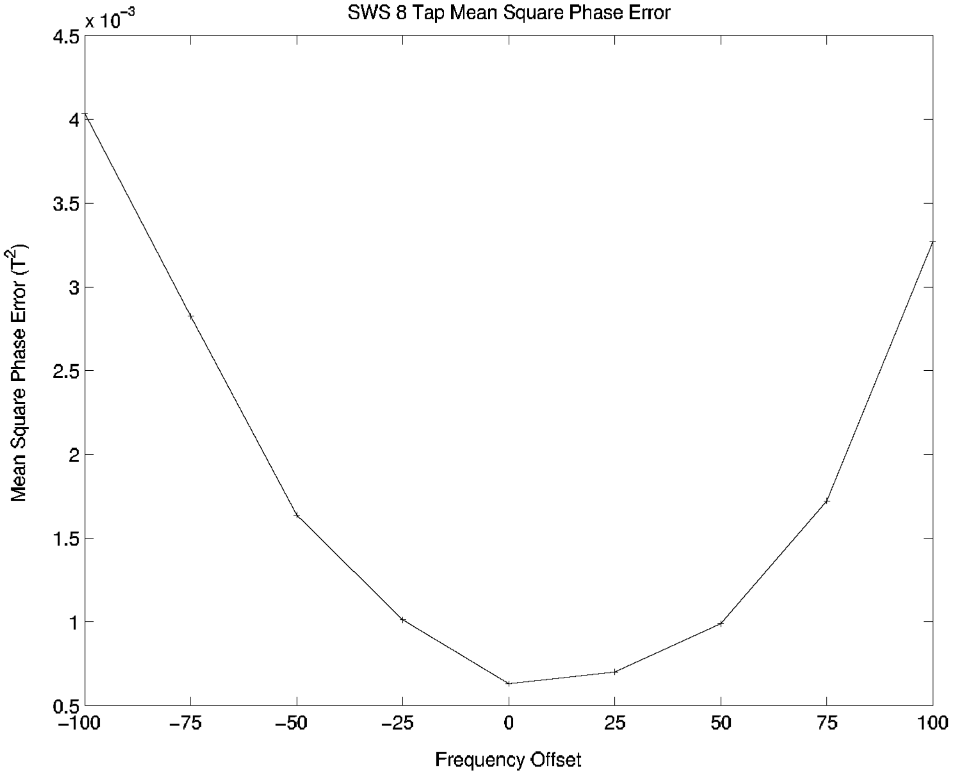


Figure 4.10: High-sample-rate SWS Frequency Offset vs. Mean-Square Phase Error Plot

Since synchronization is lost, the equivalent reduction SNR is approximately 8 dB for frequency offsets greater in magnitude than 50 ppm at an operational SNR of 10 dB. The SWS is incapable of operating in communications systems with a frequency offset on the order of 100 ppm .

The low-sample-rate SWS is also incapable of maintaining lock over the entire range of frequency offsets from -100 to +100 ppm . In fact, any frequency offset less than 10 ppm exhibits a residual phase offset and offsets greater than 10 ppm result in no synchronization.

The SWS is unable to maintain lock in initial frequency offset simulations performed at its lowest SNR for locking, 3 dB. Simulations attempted over the range of frequency offsets from -100 to +100 ppm resulted in a loss of synchronization in both the high-sample-rate and low-sample-rate versions. This indicates that the SWS will not be usable in real communications systems at its lowest SNR for locking.

4.3.5 Midstream Frequency Offset Performance

SWS's performance in the event of a midstream frequency offset, like its initial frequency offset performance, is rather poor. Figure 4.11 shows the convergence plot of the high-sample-rate SWS with an initial offset of 0.3 symbol periods and a SNR of 10 dB. The midstream frequency offset occurs at symbol 2000 and is an offset of –100 ppm . It can clearly be seen that the high-ample-rate SWS is unable to maintain synchronization following the midstream frequency offset. The high-sample-rate SWS requires the midstream frequency offset to be less than 50 ppm in order to maintain synchronization lock. This could either eliminate its practicality in some communications systems or require more accurate oscillators.

The low-sample-rate version of the SWS is unable to maintain lock after a midstream frequency offset on the order of 100 ppm . Figure 4.12 shows the convergence plot of the low-sample-rate SWS. It can be seen that once the midstream frequency offset occurs at symbol 2000, the low-sample-rate SWS is unable to maintain synchronization. Midstream frequency offsets as low as 10 ppm also exhibited the same loss of synchronization. It can be concluded from these results that the low-sample-rate SWS will not perform in real communications systems that suffer from occasional midstream frequency offsets.

4.3.5 Synchronizer Complexity

SWS is the most complex scheme studied in this work. It has the longest simulation run-time and requires the greatest memory allocation for all of its SWA. These characteristics likely would prevent it from being used in practical systems.

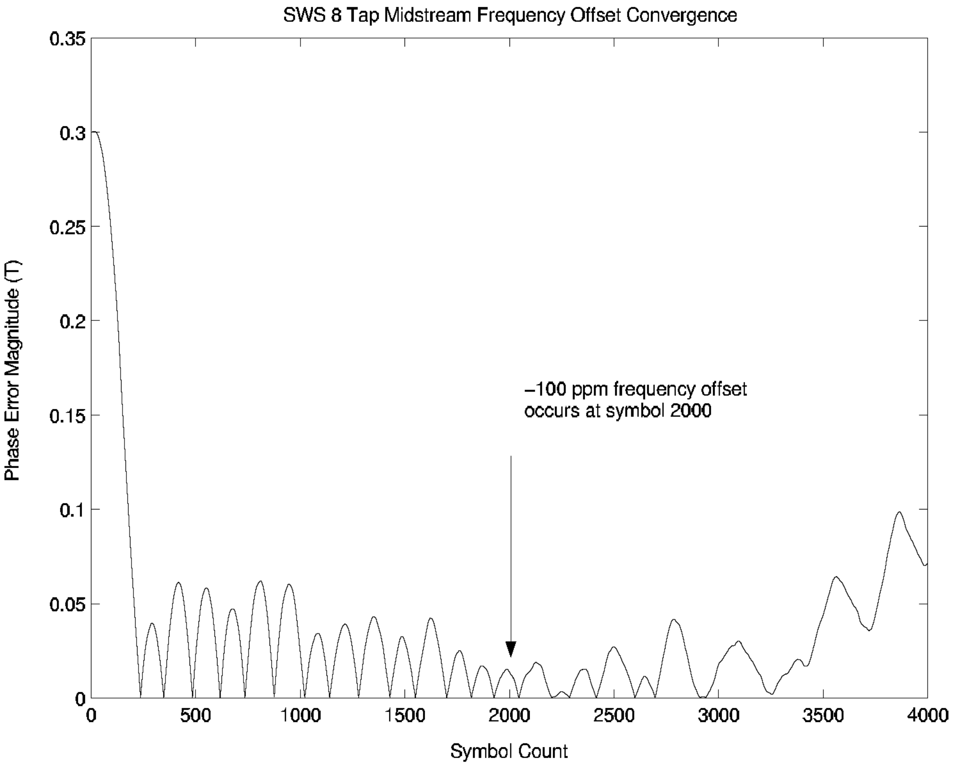


Figure 4.11: High-sample-rate SWS Midstream Frequency Offset Convergence Plot

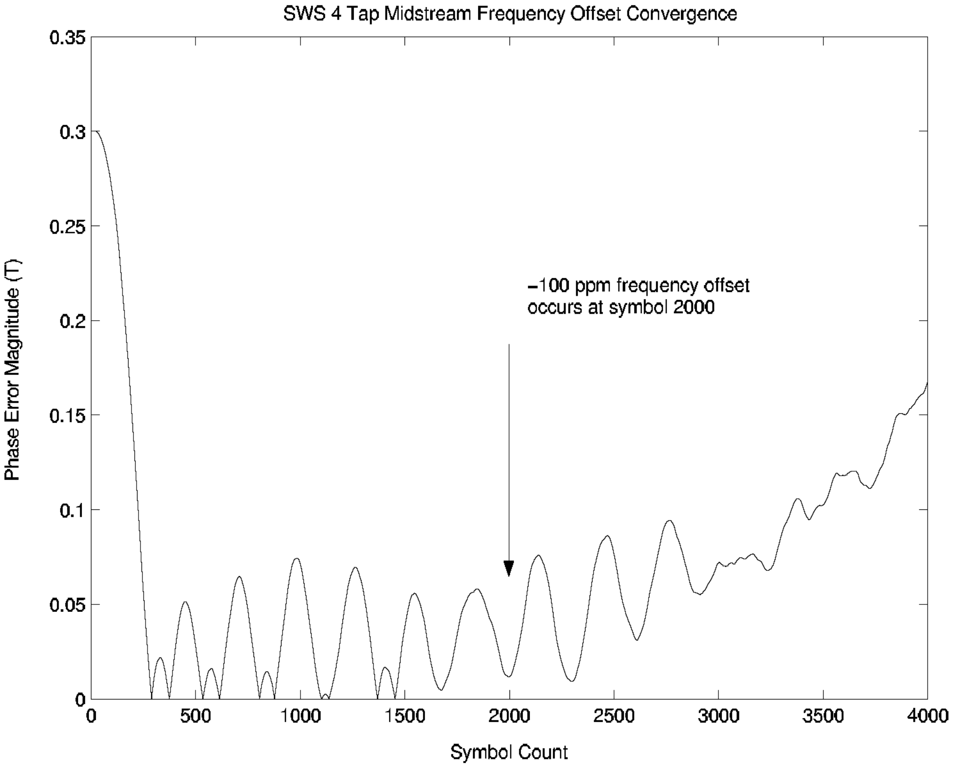


Figure 4.12: Low-sample-rate SWS Midstream Frequency Offset Convergence Plot

4.4 Modified Sliding Window Synchronizer Performance

4.4.1 Convergence Time

The convergence time of the MSWS is very short. Due to the coarse adjustment, the MSWS is able to achieve less than one tap phase error in ten symbol periods. Figure 4.13 shows the 10 dB SNR convergence plot over the first fifteen symbol periods. The initial offsets are from zero to one symbol periods in steps of 0.2 symbol periods. Note that each of the individual convergence plots achieves less than one tap phase error within the first ten symbol periods. The MSWS does have a longer period where it asymptotically approaches zero phase error. The high-sample-rate version of the MSWS takes approximately 150 symbol periods at a SNR of 10 dB to achieve near zero phase error. A plot of the convergence time of the high-sample-rate MSWS is shown in Figure 4.14. The initial offsets are from zero to one symbol periods in steps of 0.2 symbol periods. The low-sample-rate version of the MSWS also requires about 150 symbol periods at a SNR of 10 dB to achieve near zero phase error (Figure 4.15). The initial offsets are from zero to one symbol periods in steps of 0.2 symbol periods for this plot.

4.4.2 Mean-Square Phase Error

Simulation results show that the mean-square phase error of the MSWS is very low. The simulation results for SNRs in the range of 0 dB to 10 dB for the high-sample-rate version are shown in Figure 4.16. The mean-square phase error is as low as 1.13×10^{-5} T^2 at 10 dB and never greater than 9.19×10^{-5} T^2 at 0 dB. Figure 4.17 shows the mean-square phase error for the low-sample-rate MSWS over a range of SNRs from 0 dB to 10 dB. The mean-square phase error is as low as 1.48×10^{-5} T^2 at 10 dB and never greater than 9.67×10^{-5} T^2 at 0 dB.

4.4.3 Lowest SNR for Locking

The simulated performance shows that the MSWS can perform in an AWGN channel at SNR as low as 0 dB. The high-sample-rate MSWS is able to operate as low as -3 dB (E_b/N_o) with a phase error which approaches zero. It is also able to maintain synchronization lock as low as -4 dB. However, at SNRs less than -3 dB there is an increasing chance that synchronization lock will be lost. Figure 4.18 shows a phase error plot for the high-sample-rate MSWS at SNRs of -3 dB and -5 dB. The initial offset is set to 0.3 symbol periods.

The low-sample-rate version of the MSWS also performs very well. It is able to operate at SNRs as low as 0 dB. At SNRs less than 0 dB it is unable to maintain synchronization lock. Figure 4.19 shows the phase error plot for the low-sample-rate version of the MSWS at SNRs of 0 dB and -3 dB. The initial offset is set to 0.3 symbol periods.

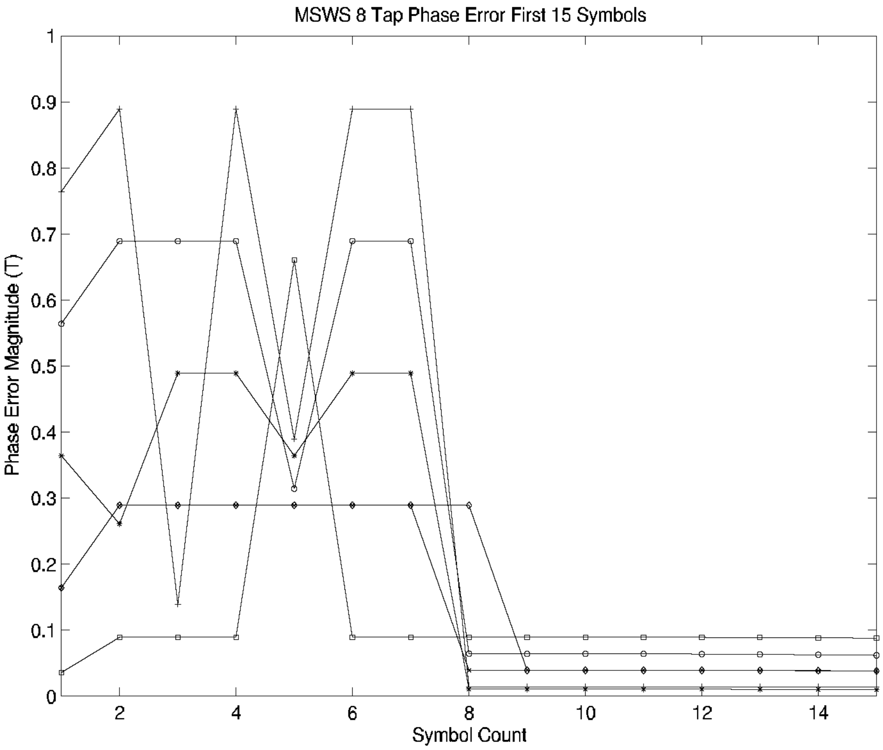


Figure 4.13: High-Sample Rate MSWS Convergence Plot First 15 Symbols

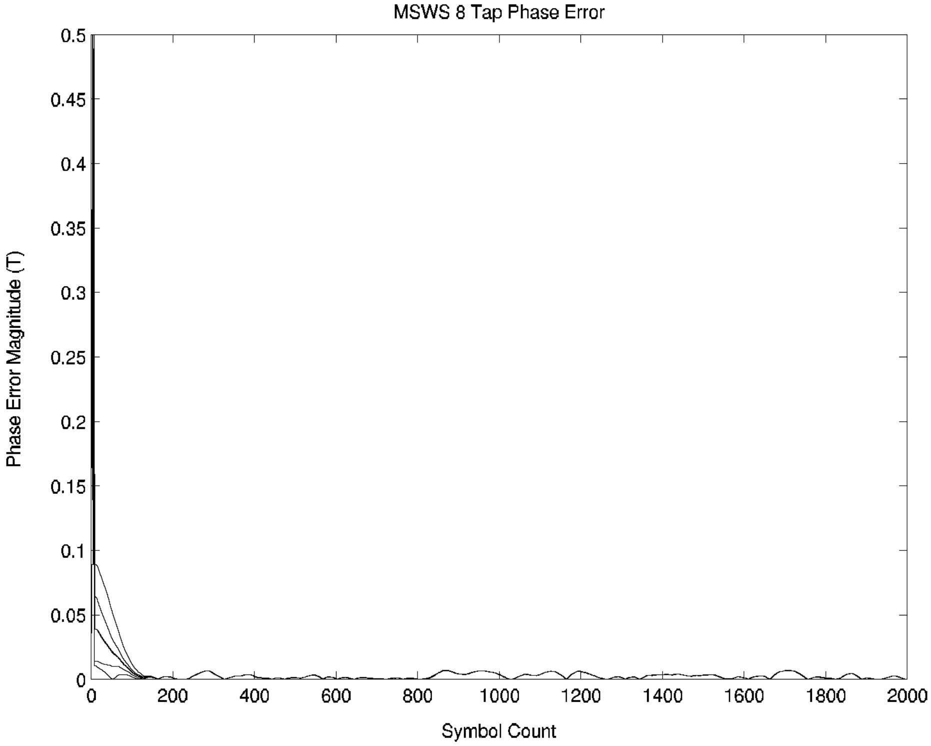


Figure 4.14: High-Sample Rate MSWS Convergence Plot

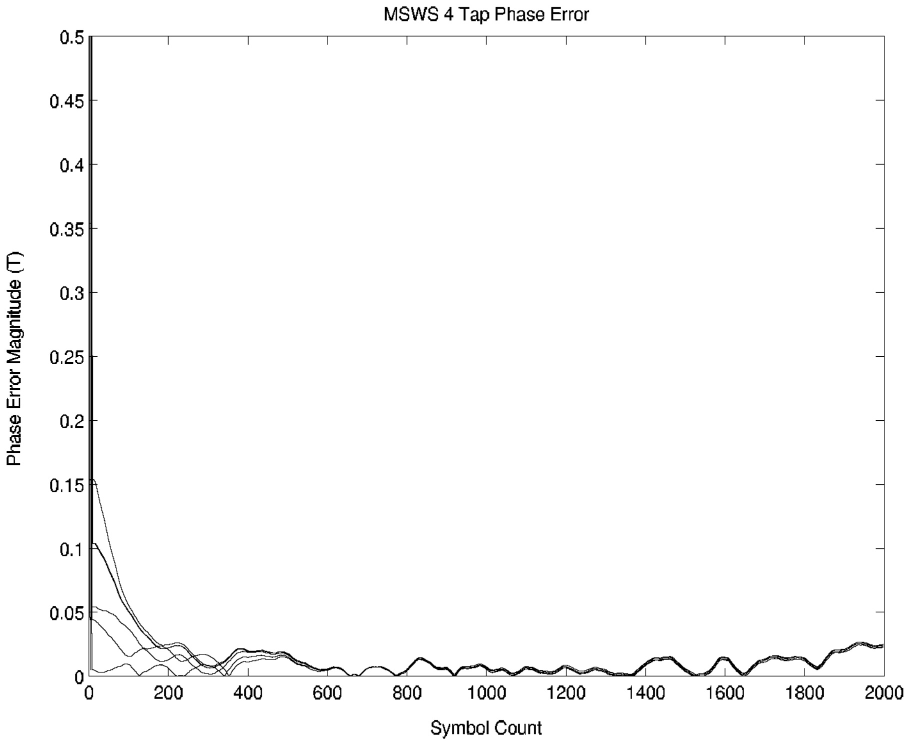


Figure 4.15: Low-Sample Rate MSWS Convergence Plot

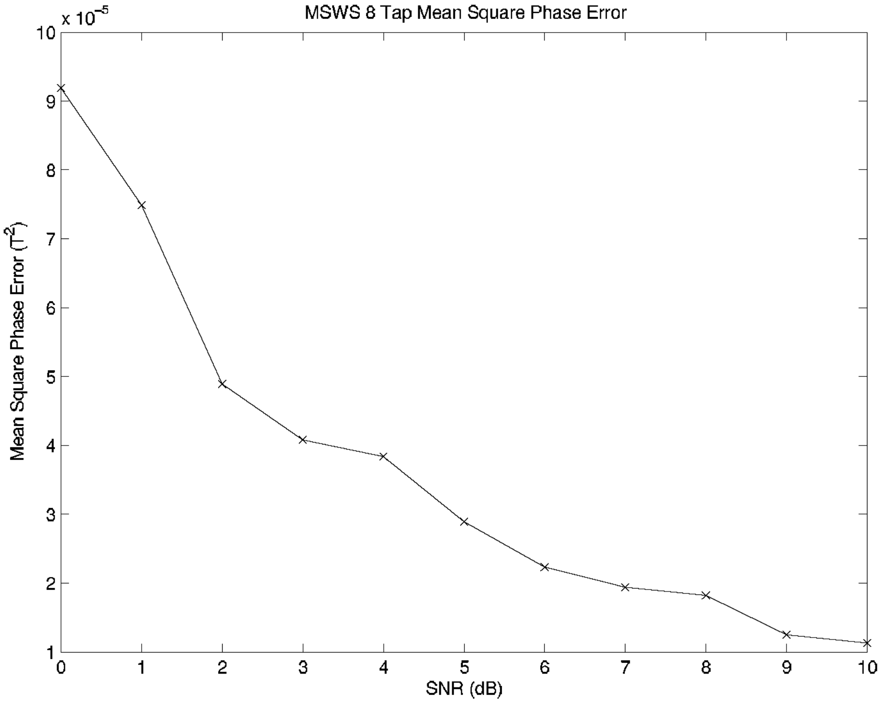


Figure 4.16: High-Sample Rate MSWS Mean-Square Phase Error Plot

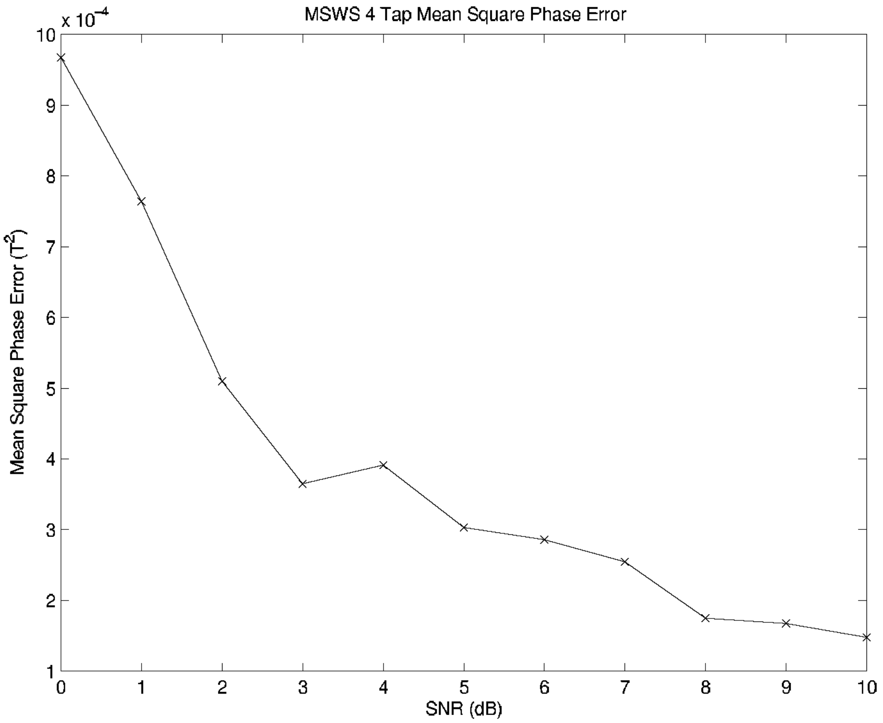


Figure 4.17: Low-Sample Rate MSWS Mean-Square Phase Error Plot

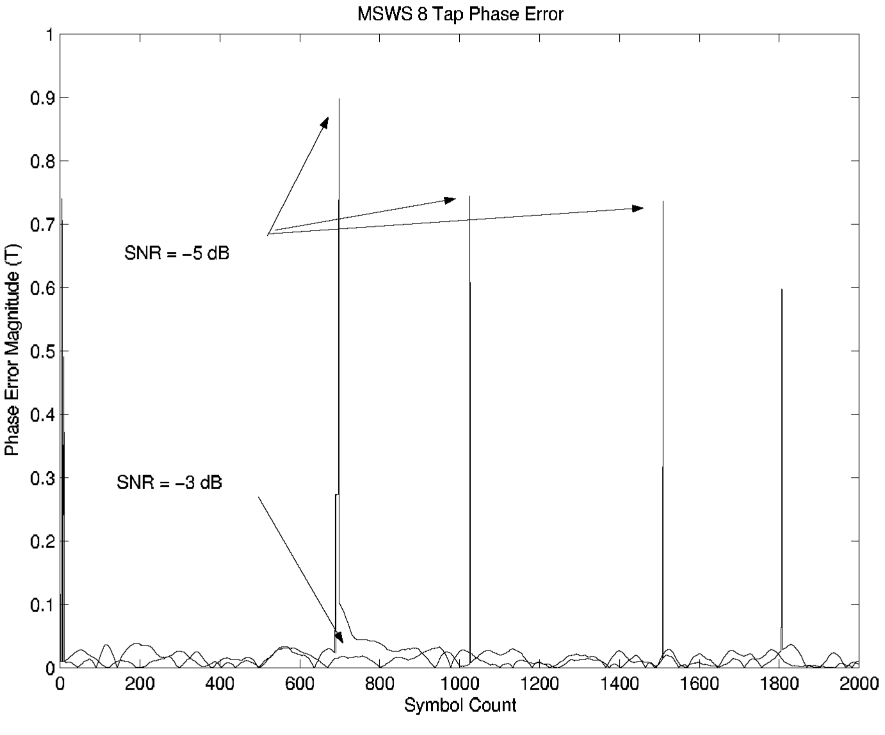


Figure 4.18: High-Sample Rate MSWS Low SNR Plot

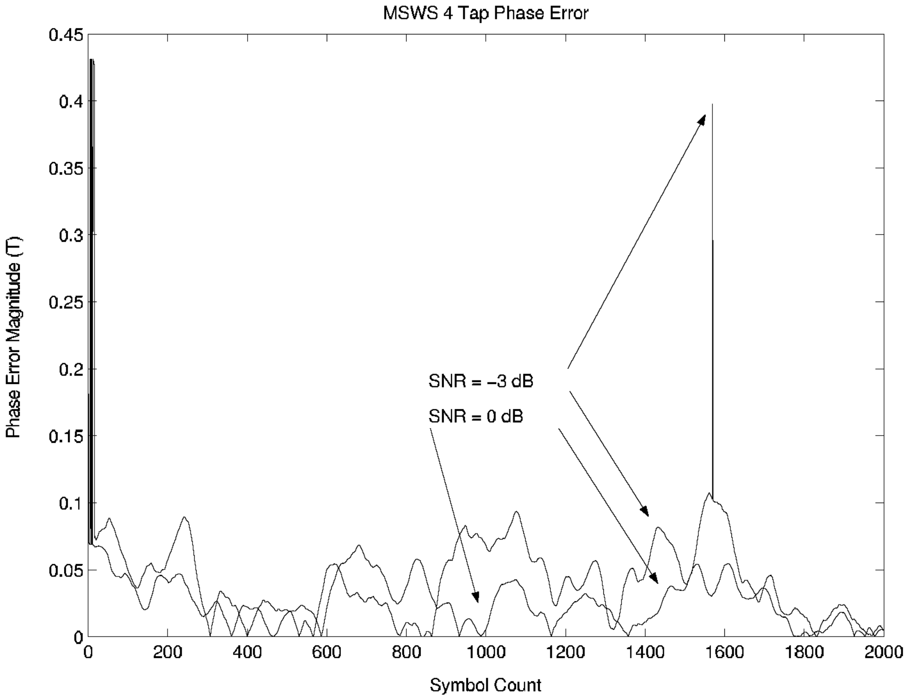


Figure 4.19: Low-Sample Rate MSWS Low SNR Plot

4.4.4 Frequency Offset Performance

The MSWS is able to achieve synchronization over the range of frequency offsets from -100 to +100 ppm. The phase error plots at a SNR of 10 dB show only minor differences over the range of frequency offsets. Figure 4.20 shows the mean-square phase error performance of the high-sample-rate MSWS with respect to initial frequency offset. The simulations were performed at a SNR of 10 dB. Comparing the worst case mean-square phase error of 3.89×10^{-5} T^2 at +100 ppm to Figure 4.16, indicates that an offset of +100 ppm is equivalent to a reduction in SNR of 6.25 dB. This indicates that the high-sample-rate MSWS will operate under frequency-offset conditions, but at a lower performance level.

The mean-square phase error vs. frequency offset plot for the low-sample-rate version of the MSWS is shown in Figure 4.21. Like the high-sample-rate version, the frequency offsets were over the range of -100 to +100 ppm and with a SNR of 10 dB. Comparing the worst-case mean-square phase error of 6.39×10^{-4} T^2 at -100 ppm to Figure 4.17 indicates that this is equivalent to a reduction in SNR of approximately 8.5 dB. If the frequency offset is kept within ±50 ppm, the equivalent degradation in SNR is less than 3 dB. This indicates that the performance of low-sample-rate version of the MSWS is highly influenced by frequency offsets and as a result, may require the use of more accurate oscillators than the other techniques.

Simulations performed at the MSWS lowest SNR for locking indicate that both high- and low-sample-rate versions can obtain and maintain synchronization lock despite frequency offsets ranging from -100 to +100 ppm. Figures 4.22 and 4.23 show plots of the initial frequency offset vs. mean-square phase error for the high- and low-sample-rate MSWS for different SNRs. The plots indicate that there is substantial performance degradation at the lowest SNR for locking. However, convergence plots showed that both high and low-sample-rate versions are able to acquire and maintain lock in these adverse conditions over the entire offset range.

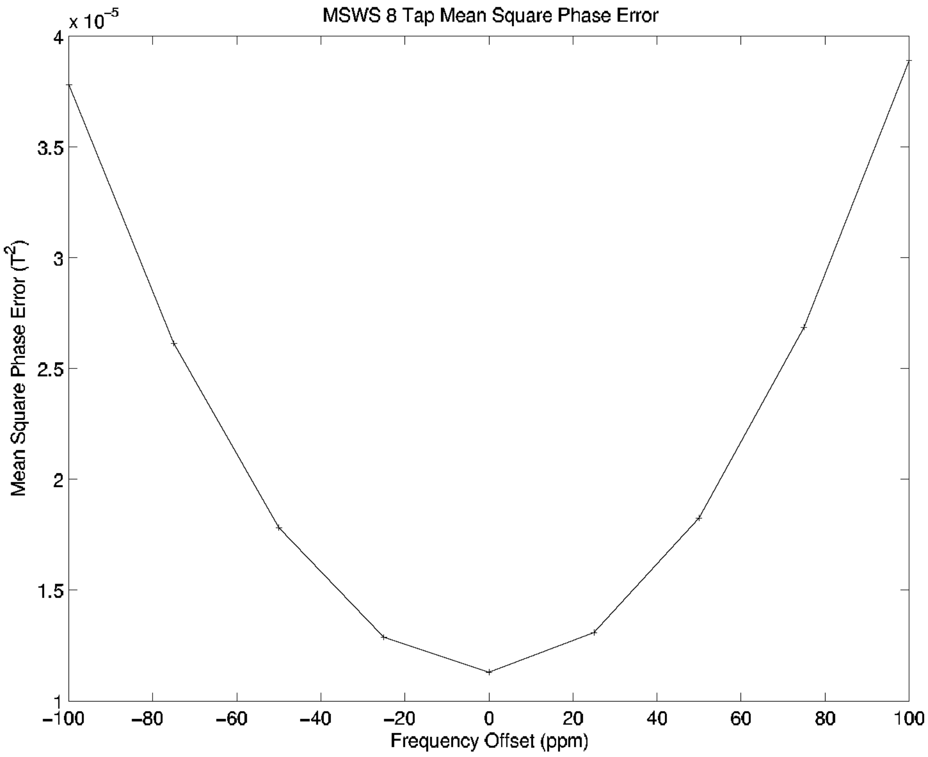


Figure 4.20: High-Sample Rate MSWS Frequency Offset vs. Mean-Square Phase Error Plot

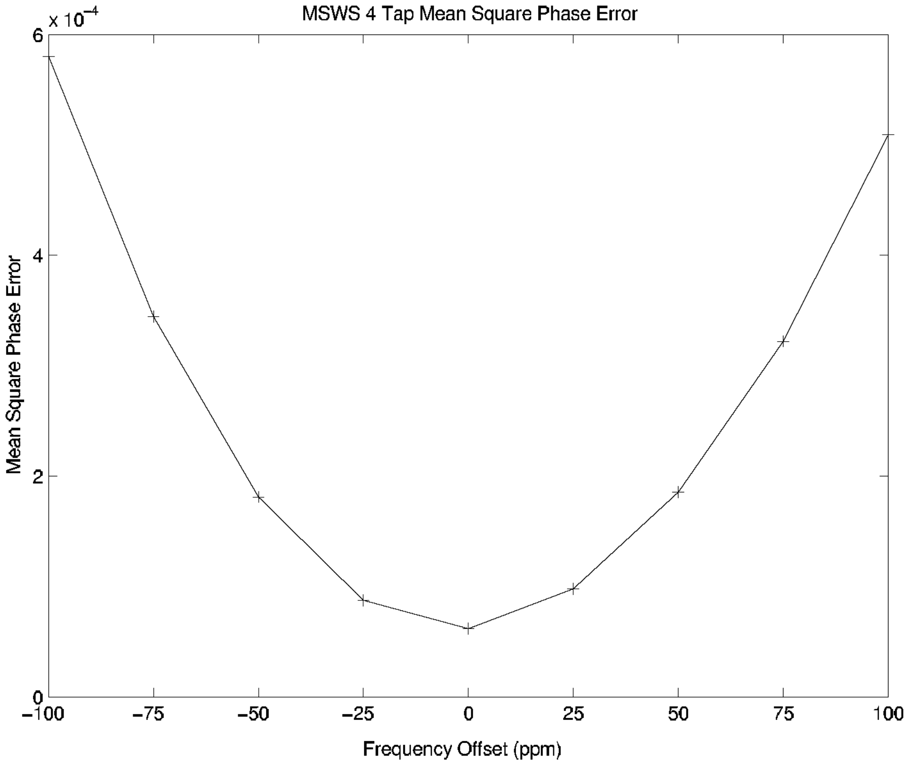


Figure 4.21: Low-Sample Rate MSWS Frequency Offset vs. Mean-Square Phase Error Plot

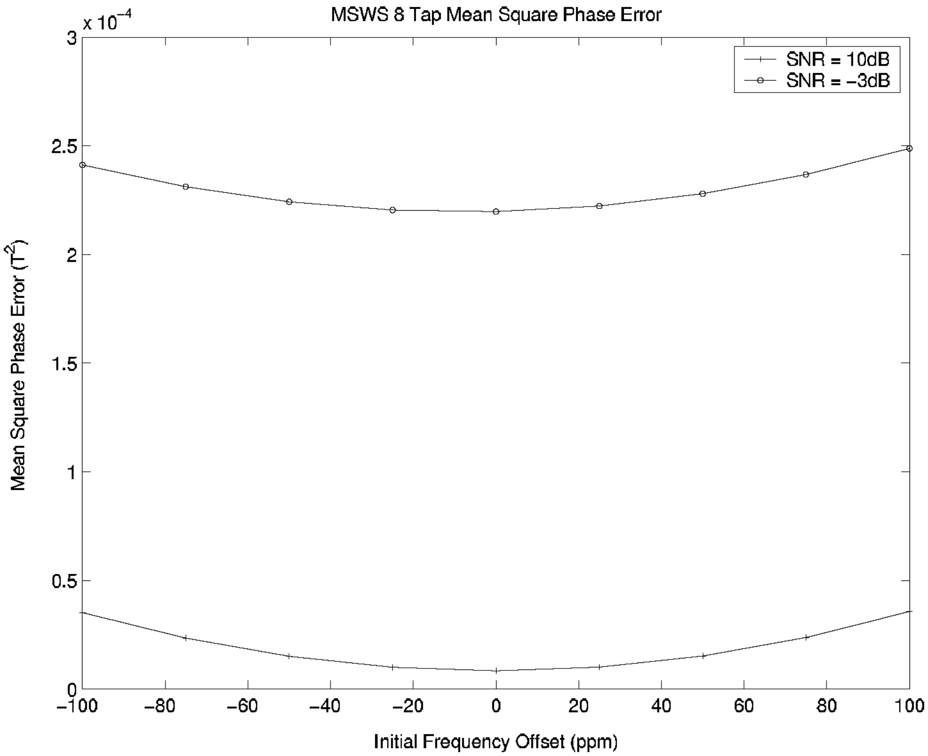


Figure 4.22: High-Sample Rate MSWS Frequency Offset vs. Mean-Square Phase Error Comparison Plot

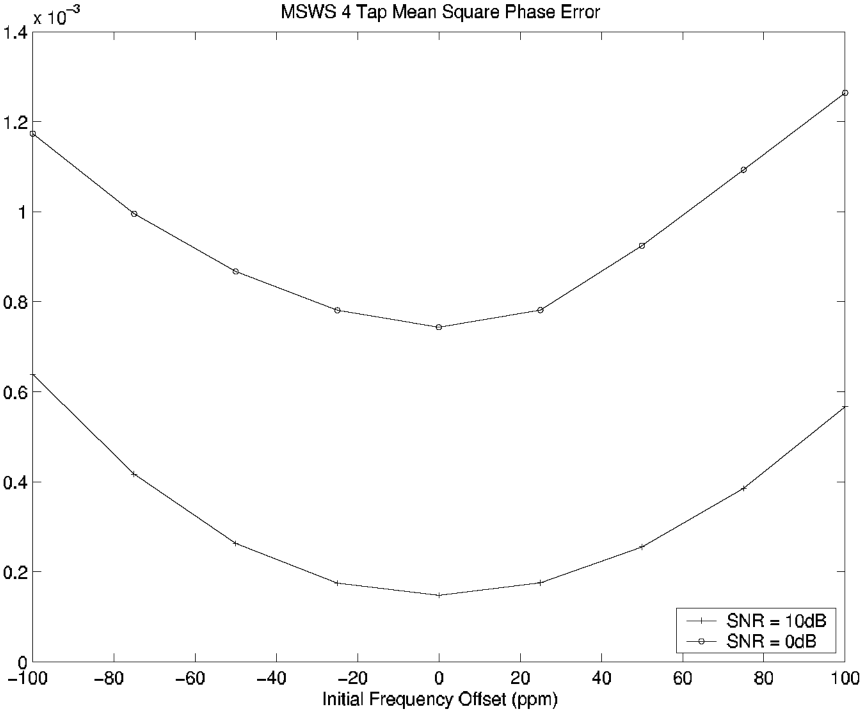


Figure 4.23: Low-Sample Rate MSWS Frequency Offset vs. Mean-Square Phase Error Comparison Plot

4.4.5 Midstream Frequency Offset Performance

The high-sample-rate MSWS is able to maintain synchronization in the event of a midstream frequency offset on the order of ±100 ppm. Figure 4.24 shows the convergence plot of the high-sample-rate MSWS with an initial offset of 0.3 symbol periods and a SNR of 10 dB. A midstream frequency offset of -100 ppm occurs at symbol 2000. The high-sample-rate MSWS is able to maintain synchronization despite the midstream frequency offset. However, the mean-square phase error jumps from 1.1×10^{-5} T^2, before the midstream frequency offset to 3.15×10^{-5} T^2 after the offset. Comparing this to the mean-square phase error plot in Figure 4.16, a midstream frequency offset of -100 ppm is approximately equivalent to a reduction in SNR of 5.25 dB.

The low-sample-rate version of the MSWS is not able to maintain a near zero phase error in the event of a midstream frequency offset on the order of ±100 ppm. Figure 4.25 shows the convergence plot of the low-sample-rate MSWS. After the -100 ppm frequency offset occurs at symbol 2000, the low-sample-rate MSWS loses the ability to maintain near zero phase error. This phenomenon does not present itself at midstream frequency offsets less than ±50 ppm. This indicates that the low-sample-rate MSWS may not be capable of operating in some communications systems and may need more accurate oscillators in both the transmitter and receiver.

4.4.6 Synchronizer Complexity

There are several primary contributors to the complexity of the MSWS including the use of a high-bandwidth, frequency-adjustable samplers, high-speed logic and bit manipulation in the sum-square calculation and control algorithm implementation, and employment of many data buffers in the creation of the time average approximation to the W(L,K) curve. It has the second longest simulation runtime of the synchronizers studied, faster only than the SWS. The system complexity is the biggest drawback, potentially limiting its applicability in low budget communications systems. Furthermore, the high-sample-rate of this scheme may preclude it from operation in high symbol rate systems. Additionally, the more stringent frequency offset requirements of the low-sample-rate MSWS will require more expensive oscillators in both the transmitter and receiver.

4.5 Early-Late Gate Synchronizer Performance

4.5.1 Convergence Time

The Early-Late Gate Synchronizer has a relatively short convergence time. It is able to achieve less than one tap phase error in about 50 symbol periods at a SNR of 10 dB. However, the ELGS is unable to obtain near zero phase error. The convergence time of the high-sample-rate ELGS is shown in Figure 4.26. The initial offsets range from zero to one symbol periods in steps of 0.2 symbol periods. The low-sample-rate ELGS convergence plot is shown in Figure 4.27. The initial offsets are also from zero to one symbol periods in steps of 0.2 symbol periods.

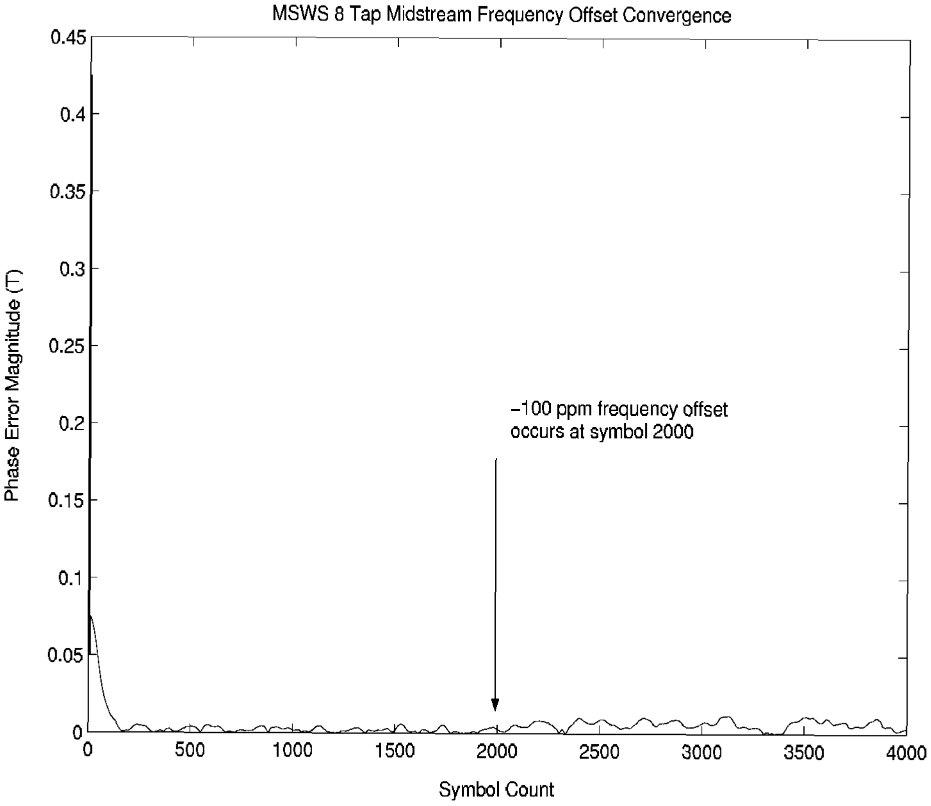


Figure 4.24: High-Sample Rate MSWS Midstream Frequency Offset Convergence Plot

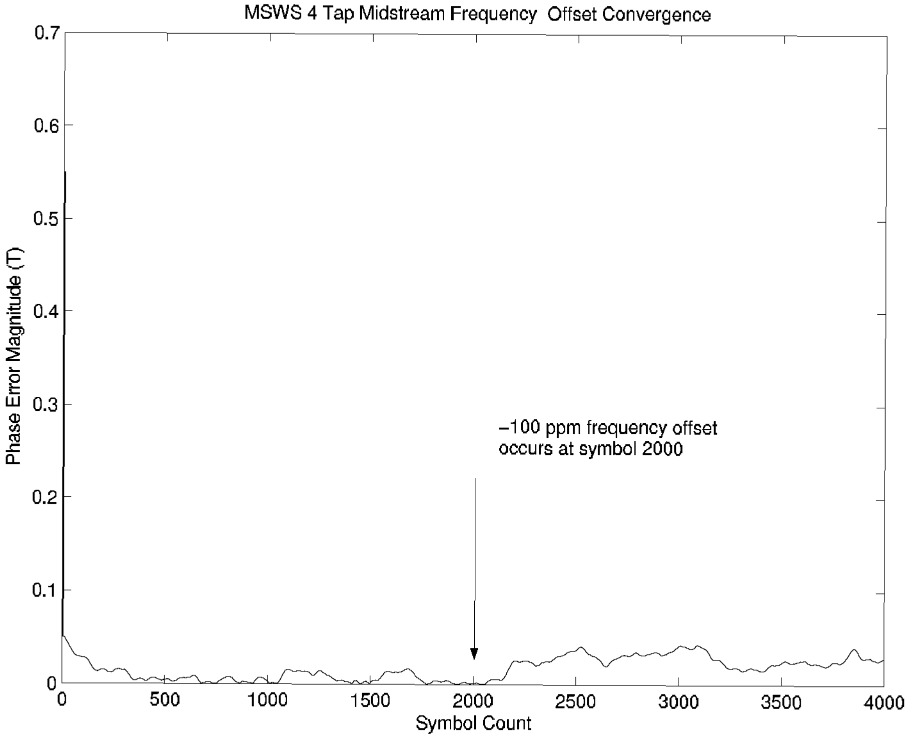


Figure 4.25: Low-Sample Rate MSWS Midstream Frequency Offset Convergence Plot

4.5.2 Mean-Square Phase Error

The mean-square phase error of the Early-Late Gate Synchronizer was not expected to be very good. The mean-square-phase error for the high-sample-rate ELGS is measured over a range of SNRs from 0 dB to 10 dB (Figure 4.28). Note that the mean-square phase error is directly correlated with the SNR at which it is operated. The lowest mean-square phase error is 1.5×10^{-4} T^2 at 10 dB and in the worst case it is 8.6×10^{-4} T^2 at 3 dB. The low-sample-rate ELGS has a lowest mean-square phase error of 1.7×10^{-3} T^2 at 10 dB and a worst case mean-square phase error of 4.2×10^{-3} T^2 at 5 dB. Figure 4.29 shows the mean-square phase error of the low-sample-rate ELGS. Again, the strong correlation between the SNR and the mean-square phase error is evident.

4.5.3 Lowest SNR for Locking

The ELGS performs rather well in terms of lowest SNR for locking. The high-sample-rate version of the ELGS is able to operate at SNRs as low as 3 dB (E_b/N_o). The low-sample-rate version is able to obtain synchronization lock at SNRs as low as 4 dB. However, neither version is able to achieve zero phase error. Figure 4.30 shows the phase error plot for the high-sample-rate ELGS. Figure 4.31 shows the phase error plot for the low-sample-rate ELGS.

4.5.4 Frequency Offset Performance

The ELGS is capable of maintaining synchronization lock over the range of frequency offsets of -100 to +100 ppm. Phase error plots at a SNR of 10 dB show no discernible differences over the range of offsets. Figure 4.32 shows the mean-square phase error performance of the high-sample-rate ELGS over the frequency offset range. The plot data is collected from simulations at SNR of 10 dB. Comparing the worst case mean-square phase error of 1.67×10^{-4} T^2 at -100 ppm to Figure 4.28 shows that an offset of -100 ppm is equivalent to a reduction in SNR of less than 0.1 dB. This indicates that the high-sample-rate ELGS will perform well under frequency offset conditions.

The low-sample-rate version of the ELGS also performs well in the event of a frequency offset. Over the range of frequency offsets from -100 to +100 PPM at a SNR of 10 dB, the phase error plots of the low-sample-rate ELGS are almost identical. Figure 4.33 shows the mean-square phase error of the low-sample-rate ELGS over the frequency offset range. Note that the worst case mean-square-phase error for the low-sample-rate ELGS is 1.81×10^{-3} T^2 at a frequency offset of -100 ppm. Comparing this to Figure 4.29 it is clear that a frequency offset of -100 ppm is equivalent to a reduction in SNR of less than 0.1 dB.

At the lowest SNR for locking, 3 dB, the high-sample-rate ELGS is able to obtain and maintain lock at frequency offsets ranging from -100 to +100 ppm. The low-sample-rate version is also able to operate at its lowest SNR for locking, 4 dB, with frequency offsets of -100 to +100 ppm As a result, the ELGS should be capable of operating in real systems at SNRs as low as 3 dB and 4 dB, with initial frequency offsets on the order of ±100 ppm.

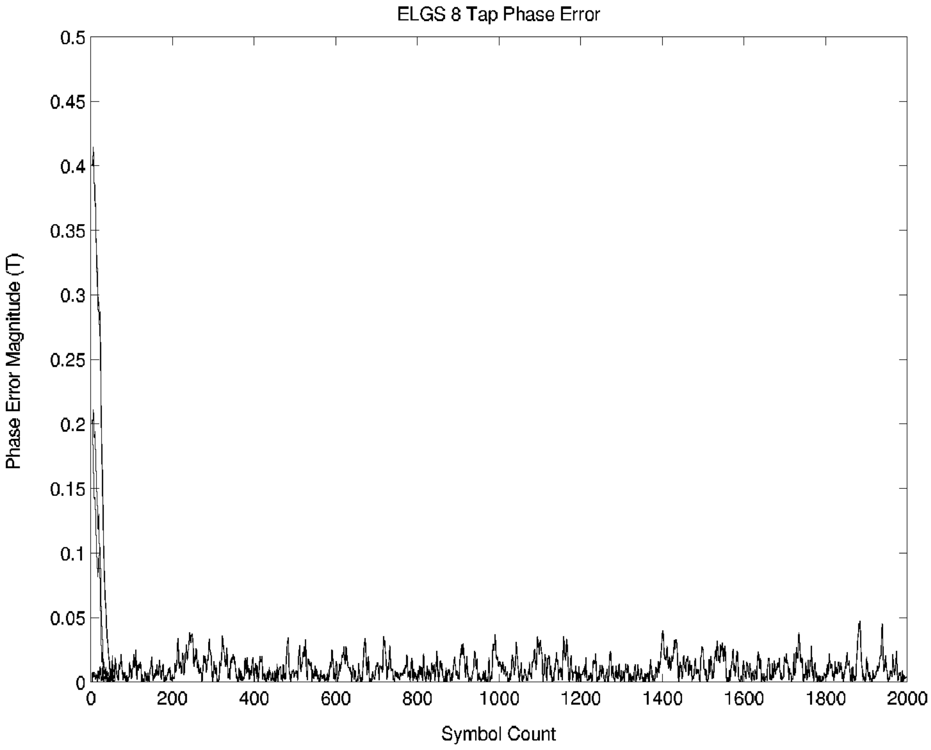


Figure 4.26: High-Sample Rate ELGS Convergence Plot

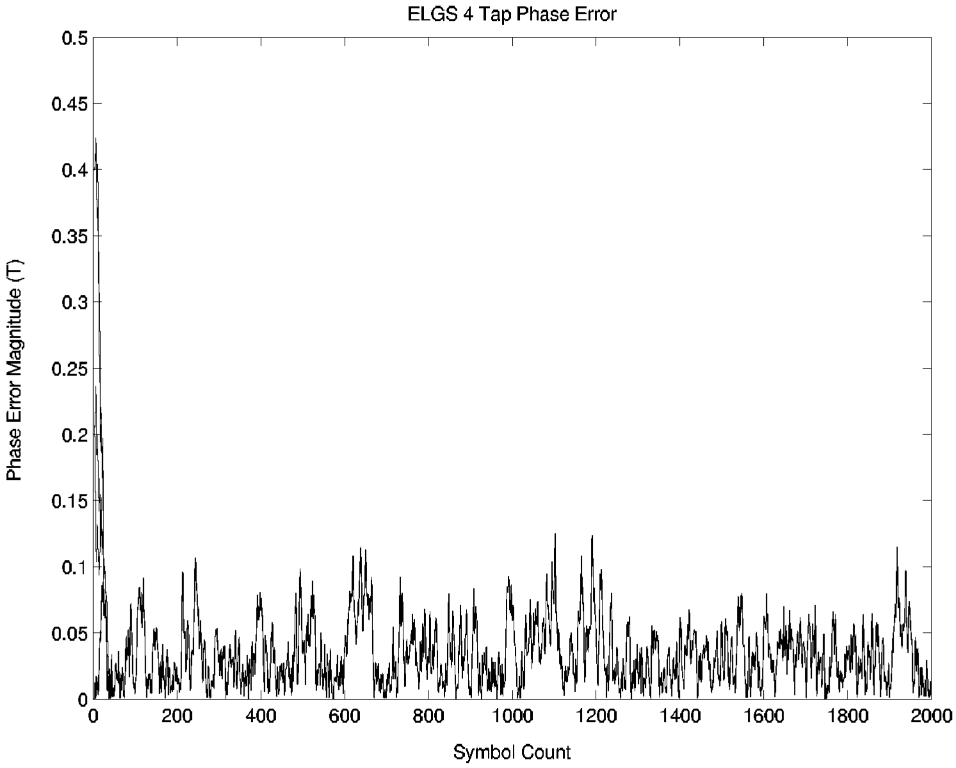


Figure 4.27: Low-Sample Rate ELGS Convergence Plot

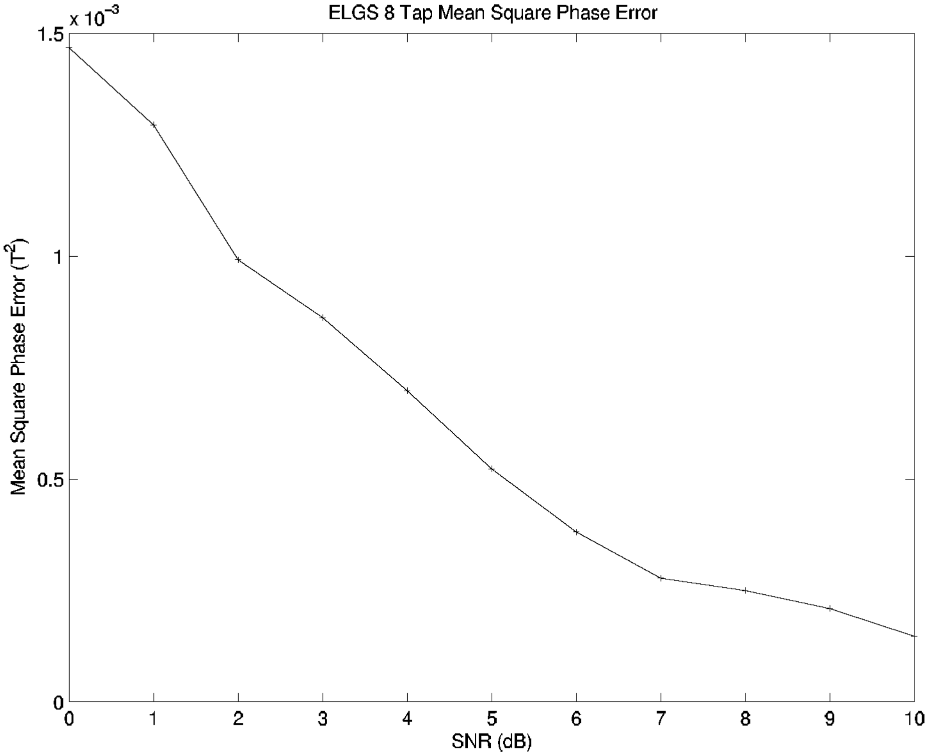


Figure 4.28: High-Sample Rate ELGS Mean-Square Phase Error Plot

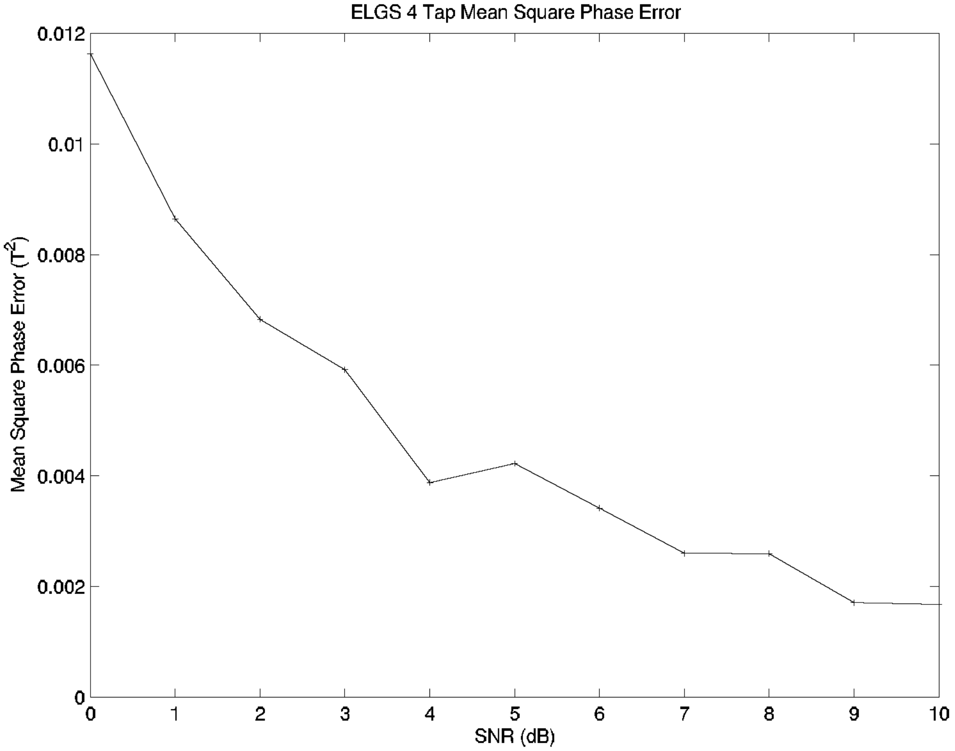


Figure 4.29: Low-Sample Rate ELGS Mean-Square Phase Error Plot

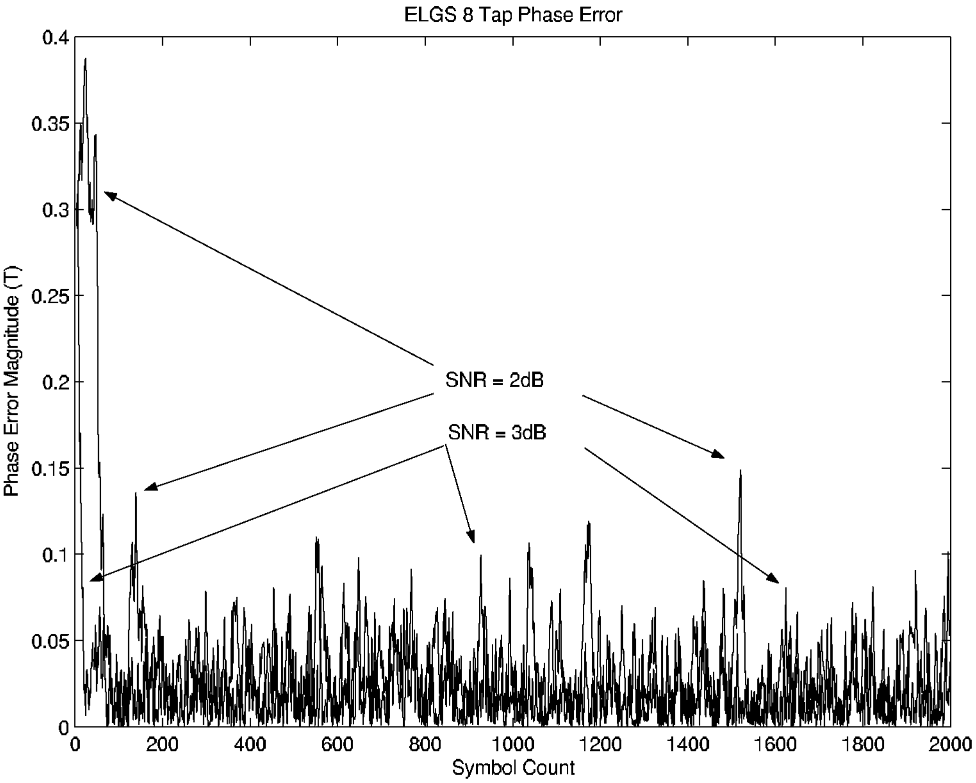


Figure 4.30: High-Sample Rate ELGS Low SNR Plot

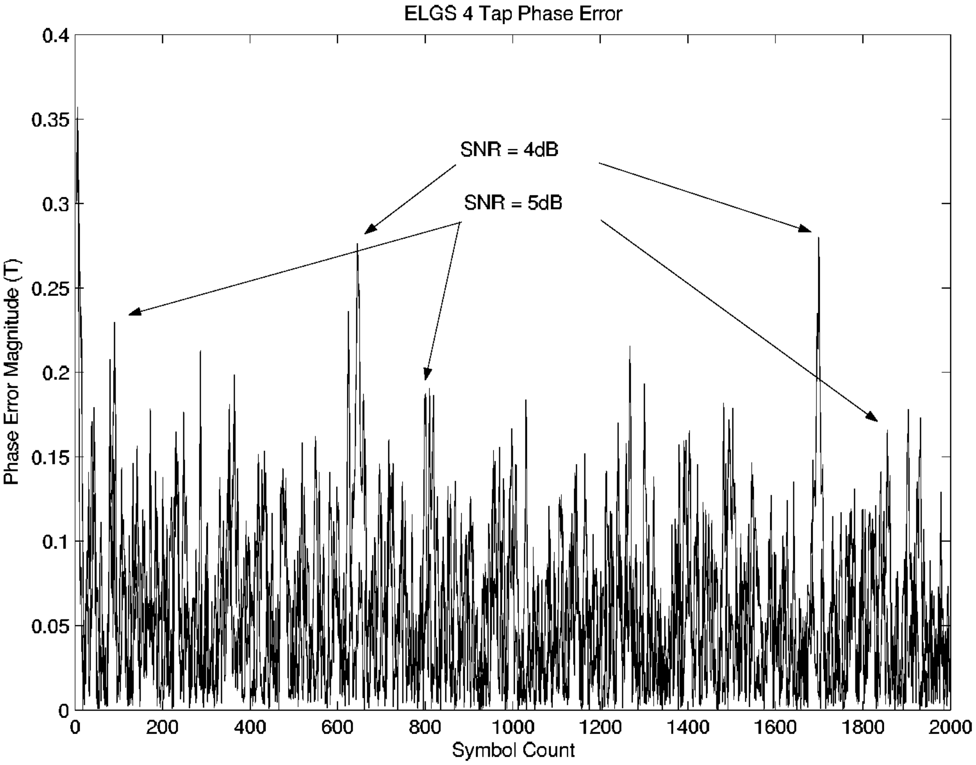


Figure 4.31: Low-Sample Rate ELGS Low SNR Plot

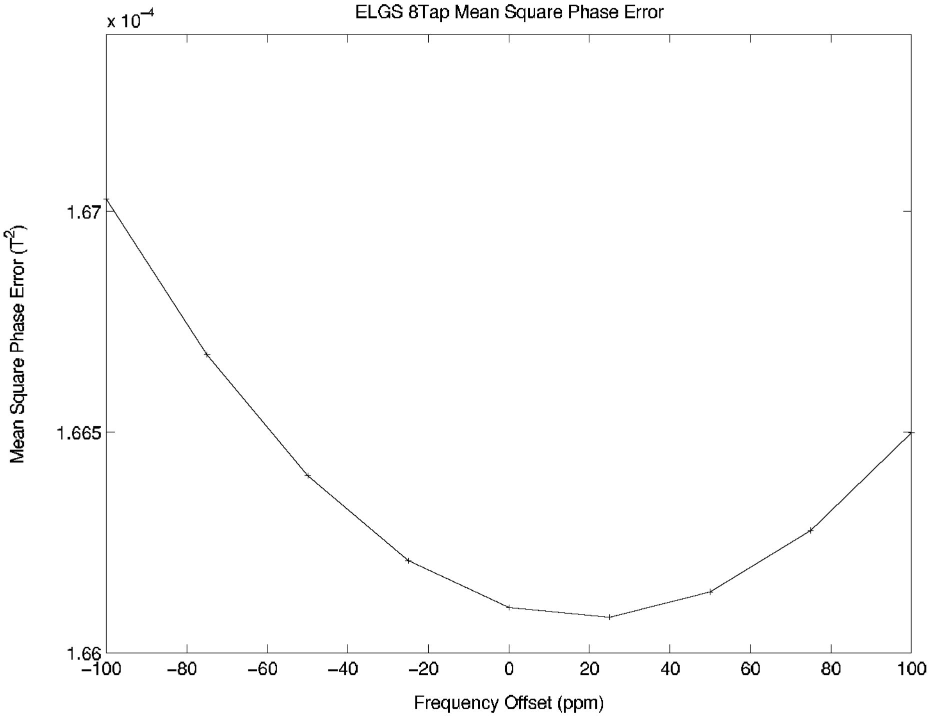


Figure 4.32: High-Sample Rate ELGS Frequency Offset vs. Mean-Square Phase Error Plot

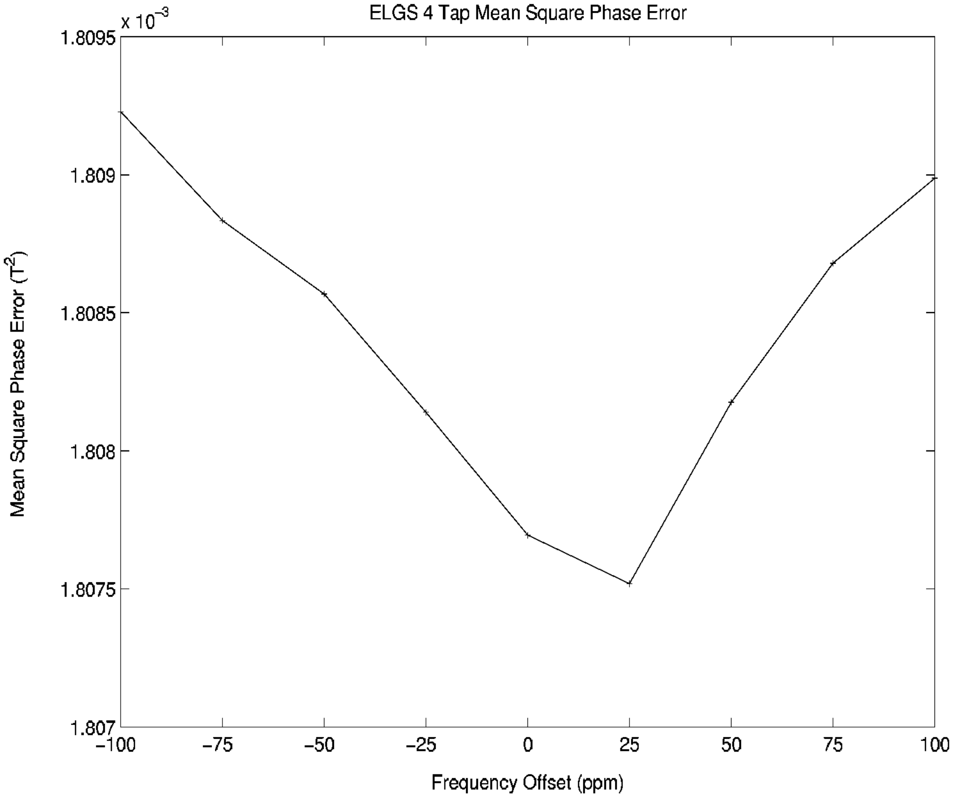


Figure 4.33: Low-Sample Rate ELGS Frequency Offset vs. Mean-Square Phase Error Plot

4.5.5 Midstream Frequency Offset Performance

The high-sample-rate ELGS is able to maintain synchronization despite midstream frequency offsets on the order of ±100 ppm. Figure 4.34 shows the convergence plot of the high-sample-rate ELGS with an initial offset of 0.3 symbol periods and a SNR of 10 dB. There is no visual difference in the convergence prior to, or following, the -100 ppm frequency offset at symbol 2000. The mean-square phase error prior to the frequency offset is 1.55×10^{-4} T^2 and 1.39×10^{-4} T^2 following the offset.

The low-sample-rate version of the ELGS is also able to maintain synchronization in the event of a midstream frequency offset on the order of ±100 ppm. Figure 4.35 shows the convergence plot of the low-sample-rate ELGS with an initial phase offset of 0.3 symbol periods and a SNR of 10 dB. Like the high-sample-rate version, the low-sample-rate ELGS exhibits no visual difference in the convergence plot prior to, or following, the midstream frequency offset of -100 ppm. The mean-square phase error prior to the frequency offset is 1.7×10^{-3} T^2 and 1.6×10^{-3} T^2 after the offset. This indicates that the low-sample-rate version of the ELGS will be able to operate in most communication systems, provided that they do not require a mean-square phase error on the order of 1.0×10^{-3} T^2 or less.

4.5.6 Synchronizer Complexity

The ELGS is a fairly simple scheme. It has the second fastest runtime, slower only than the GZCD. The simplicity of the ELGS enables it to fit a wide range of system budgets. However, its high-sample rate may limit it to low speed communication systems.

4.6 Gardner Zero-Crossing Detector Performance

4.6.1 Convergence Time

The GZCD has a fairly fast convergence time. With a SNR of 10 dB, the GZCD is able to converge in less than 30 symbol periods (Figure 3.36). The initial offsets range from zero to one symbol period in steps of 0.2 symbol periods for this plot. Note that following its fast convergence, the GZCD is unable to achieve zero phase error (Figure 3.36).

4.6.2 Mean-Square Phase Error

The mean-square phase error of GZCD is measured over the range of SNRs from 0 dB to 10 dB. In this range it is able to achieve a mean-square phase error as low as 1.3×10^{-4} T^2 at 10 dB. The maximum mean-square phase error produced in the GZCD operational range is 8.7×10^{-4} T^2 at a SNR of 3 dB. The plot of mean-square phase error versus SNR for the GZCD is shown in Figure 3.37. Note the strong correlation between the SNR and mean-square phase error.

4.6.3 Lowest SNR for Locking

The GZCD is able to operate at SNRs as low as 3 dB (E_b/N_o). At SNRs less than 3 dB, the GZCD shows a phase error greater than 0.125 symbol periods. Figure4.38 shows the lowest SNR for locking for the GZCD.

4.6.4 Frequency Offset Performance

The GZCD is able to maintain synchronization over a range of frequency offsets from -100 to +100 ppm. The phase error plots of the GZCD over this range of frequency offsets are almost indiscernible from one another. The mean-square phase error plot of the GZCD over the range of frequency offsets is shown in Figure 4.39. At a SNR of 10 dB, the worst-case mean-square phase error, 1.403×10^{-4} T^2, occurs at -100 ppm. Comparing this value with the results in Figure 4.37, one can see that a frequency offset of -100 ppm is equivalent to a reduction in SNR of less than 0.1 dB. This indicates that a frequency offset over the range of -100 to +100 ppm will not have a significant effect on the performance of the GZCD.

Simulations performed at the lowest SNR for locking, 3 dB, indicate that the GZCD is able to obtain and maintain synchronization despite frequency offsets ranging from -100 to +100 ppm. Convergence plots show no loss of synchronization as well as a quick acquisition despite these adverse conditions. As a result, the GZCD should be suitable for real communications systems which have a SNR of 3 dB or greater.

4.6.5 Midstream Frequency Performance

The midstream frequency offset convergence plot of the GZCD is shown in Figure 4.40. The initial phase offset is 0.3 symbol periods, the SNR is 10 dB, and the -100 ppm frequency offset occurs at symbol 2000. There is no visible difference in the convergence of the GZCD prior to, or following, the frequency offset. The mean-square phase error before the offset is 1.1×10^{-4} T^2 and 1.34×10^{-4} T^2 after the offset. This indicates that midstream frequency offsets on the order of ±100 ppm have only a minor effect on the performance of the GZCD.

4.6.6 Synchronizer Complexity

The GZCD is the least complex of all of the schemes studied in this analysis. Its simulation runtime is the fastest, and it operates at the lowest sample rate. Given these advantages, the GZCD is applicable to a wide range of communication systems.